



UNIVERSITÄT ZU LÜBECK  
INSTITUT FÜR MULTIMEDIALE  
UND INTERAKTIVE SYSTEME

# **Snipe-IT Companion: Entwicklung eines Reservierungstools für Assets**

Snipe-IT Companion: Development of a reservation tool for  
assets

**Bachelorarbeit**

im Rahmen des Studiengangs

**Medieninformatik**

der Universität zu Lübeck

vorgelegt von

**Anna-Tabea Manske**

ausgegeben und betreut von

**Univ.-Prof. Dr. rer. nat. Hans-Christian Jetter**

mit Unterstützung von

Jan-Henrik Schröder, M.Sc.

Lübeck, den 03. November 2022



## **Kurzfassung**

Das Ziel dieser Arbeit ist es ein System zu entwickeln, welches den Reservierungs- und Entleihprozess einheitlicher, effizienter und zufriedenstellender lösen lässt. Die Anwendung soll es unter anderem ermöglichen, die auszuleihenden Assets in einer Liste abzubilden und diese zu durchsuchen. Hierbei ist auch das Anzeigen einer Stückzahl der jeweils verfügbaren Assets, deren Bedienungsanleitung sowie der verantwortliche Mitarbeitende denkbar.

Ziel dieser Arbeit ist es, abgeleitet aus den eingangs beschriebenen Anforderungen, ein wirksames Reservierungstool für Assets zu konzipieren, implementieren und evaluieren.



## **Abstract**

The abstract should not be longer than some paragraphs. There must be an English translation of the German abstract, which has to be the exact translation of the German version.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Ziel der Arbeit . . . . .	2
1.2	Forschungsfragen . . . . .	2
1.3	Vorgehensweise . . . . .	3
<b>2</b>	<b>Anforderungen</b>	<b>5</b>
2.1	Datenquellen . . . . .	5
2.2	Benutzeranalyse . . . . .	8
2.3	Problemanalyse . . . . .	9
2.4	Aufgabenanalyse . . . . .	12
2.5	Kontextanalyse . . . . .	15
2.6	Formalisierte Anforderungen . . . . .	17
<b>3</b>	<b>Konzeption</b>	<b>23</b>
3.1	Funktionalität . . . . .	23
3.2	Systemarchitektur . . . . .	27
<b>4</b>	<b>Interface-Design</b>	<b>33</b>
4.1	Iteratives Vorgehen . . . . .	34
4.2	Designsprache . . . . .	34
<b>5</b>	<b>Implementierung</b>	<b>43</b>
5.1	Implementierung des Reservierungsinterfaces . . . . .	43
5.2	Implementierung des Frontends . . . . .	48
5.3	Nutzung des Systems . . . . .	51
5.4	Fazit der Implementierung . . . . .	52
<b>6</b>	<b>Dialogbeispiele</b>	<b>53</b>
<b>7</b>	<b>Evaluation</b>	<b>61</b>
7.1	Ziele . . . . .	61

7.2	Vorgehen und Methodik . . . . .	61
7.3	ATI und UEQ Ergebnisse . . . . .	62
7.4	Verleihende . . . . .	64
7.5	Ausleihende . . . . .	65
7.6	Diskussion . . . . .	68
<b>8</b>	<b>Zusammenfassung und Ausblick</b>	<b>69</b>
8.1	Zusammenfassung . . . . .	69
8.2	Offene Punkte . . . . .	69
8.3	Ausblick . . . . .	70
8.4	Abschließendes Fazit . . . . .	70
	<b>Abkürzungsverzeichnis</b>	<b>71</b>
	<b>Abbildungsverzeichnis</b>	<b>73</b>
	<b>Tabellenverzeichnis</b>	<b>75</b>
	<b>Quellcodeverzeichnis</b>	<b>77</b>
	<b>Literaturverzeichnis</b>	<b>79</b>
	<b>Anhang</b>	<b>83</b>
A	Interviewleitfaden der Analyse . . . . .	83
B	Die zehn Usability Heuristiken nach Jakob Nielsen . . . . .	86
C	Leitfaden und Aufgaben der Evaluation . . . . .	87
D	Umfrage der Evaluation . . . . .	90
E	Digitale Medien . . . . .	90



# 1

## Einleitung

Das Ausleihen von Assets jeglicher Art ist keine Neuheit (Söderholm, 2018). An der Zentralen Hochschulbibliothek Lübeck (ZHB) wird hierzu beispielsweise die Buchungsapp *Affluences* verwendet. Zum Ausleihen von Materialien, müssen Terminabholungen online gebucht werden (ZHB, o. J.). Die Anfragen können überprüft werden, wodurch das vorausschauende Planen der Materialien ermöglicht wird. Am Institut für Multimediale und Interaktive Systeme (IMIS) werden Assets ohne Anmeldung oder mit einer mündlichen Absprache verwendet. Aufseiten der Mitarbeitenden wird der Gebrauch der Assets individuell geplant, sodass das frühzeitige Reservieren oder das geplante Ausleihen erschwert werden. Für die vorausschauende Planung von anstehenden Projekten gibt es keine feste Reservierung oder einen Überblick, wann die gewünschten Assets wieder verfügbar sind. Zudem ist vielen Mitarbeitenden unklar, welche Assets sich in den Laboren des IMIS befinden. Folglich kennen Mitarbeitende nur selten die Möglichkeiten, mit denen sie ihre Forschungsprojekte oder die Lehre ergänzen könnten.

Aktuell werden Reservierungen sowie der Gebrauch von Equipment über unterschiedliche Kommunikationswege wie E-Mail oder mündlicher Absprache bei zuständigen Mitarbeitenden angefragt. Die zuständigen Mitarbeitenden prüfen die Anfrage und koordinieren potenzielle Kollisionen mit bereits reservierten Zeiträumen oder Absprachen und bestätigen, ändern oder lehnen diese (Reservierungs-)Anfrage ab. In einigen Fällen werden die gebuchten Zeiten auf Papier dokumentiert. Aufseiten der Studierenden kann ein solches System für Gruppenarbeiten, wie zum Beispiel die des Moduls „Einführung in die Medieninformatik“ hilfreich sein. Studierende müssen während dieses Projekts eine multimediale Abgabe produzieren, welche beispielsweise die Form eines Videos haben kann. Aktuell ist vielen Studierenden nicht bewusst, dass das IMIS Equipment wie Kameras, Greenscreens oder Gimbals für Videos bereitstellt. Des Weiteren sind Hemmschwelle und Aufwand zum Ausleihen der Hardware hoch, da uneindeutig ist, welche Mitarbeitenden für die jeweilige Hardware zuständig sind.

Formulierung

### 1.1 Ziel der Arbeit

Das Ziel dieser Arbeit ist es ein wirksames System zu entwickeln, welches den Reservierungs- und Ausleihprozess am IMIS einheitlicher, effizienter und zufriedenstellender lösen lässt. Die Anwendung soll es unter anderem ermöglichen, die auszuleihenden Assets in einer Liste abzubilden und diese zu durchsuchen. In diesem Zusammenhang soll auch das Anzeigen einer Stückzahl der jeweils verfügbaren Assets, deren Bedienungsanleitung sowie die verantwortlichen Mitarbeitenden umgesetzt werden. Das Reservierungstool soll eine niedrigschwellige Möglichkeit bieten, um Mitarbeitenden die Arbeit im Reservierungsprozess zu erleichtern. Außerdem soll es eine Übersicht über Assets geben, sodass Mitarbeitende nicht mehr im Unklaren über die Hardware-Möglichkeiten sind und das Material optimal genutzt werden kann.

Die Basis für ein solches Tool schafft die Asset-Managementsoftware *Snipe-IT* (Snipe-IT, o. J.), welche bereits am IMIS eingesetzt wird. *Snipe-IT* ist eine kostenlose, quelloffene IT-Asset-Verwaltungs-Plattform, welche das Nachverfolgen von Software-Lizenzen, Hardware und Verbrauchsgegenständen ermöglicht. Genannte Assets können über ein Dashboard hinzugefügt, verwaltet und gelöscht werden. Über Labels können Assets zur Übersichtlichkeit in verschiedene Kategorien eingeordnet werden, während Tags ein Asset eindeutig identifizieren (z. B. Seriennummer). Zudem ermöglicht das „Checkin/Checkout“-System die Nachverfolgung aller Assets, falls diese zum Beispiel an eine Person ausgeliehen werden. Zu jedem Zeitpunkt kann ein Asset maximal einer Person zugeordnet werden, wodurch das mehrfache gleichzeitige Ausleihen eines Assets verhindert wird. Da *Snipe-IT* selbst die zukünftige Reservierung nicht unterstützt, umfasst das Ziel der Arbeit einen *Snipe-IT Companion*, welcher das Ausleihen in die Zukunft möglich macht.

### 1.2 Forschungsfragen

Im Sinne der eingangs beschriebenen Ziele soll im Rahmen der vorliegenden Arbeit untersucht werden, wie ein Reservierungstool gestaltet werden kann, um Mitarbeitende und Studierende darin zu unterstützen, Assets einzusehen und ausleihen zu können.

Um den aktuellen Stand und die Probleme des aktuellen Ausleihprozess nachvollziehen zu können, müssen diese zunächst ermittelt und klassifiziert werden. Die erste Forschungsfrage beschäftigt sich daher mit der Analyse der Schwierigkeit des Ausleihprozesses.

- F1 | Welche zentralen Schwierigkeiten bringt die aktuelle Planung und Reservierung von Assets für Mitarbeitende und Studierende mit sich?

Um die zentralen Schwierigkeiten und Probleme lösen zu können, müssen Anforderungen an das System ermittelt werden. Anschließend müssen diese nach Relevanz sortiert werden, um möglichst viele Schwierigkeiten lösen zu können. Die zweite Forschungsfrage beschäftigt sich folglich mit den Anforderungen, welche ein solches Reservierungstool umfassen sollte, um die Schwierigkeiten zu minimieren.

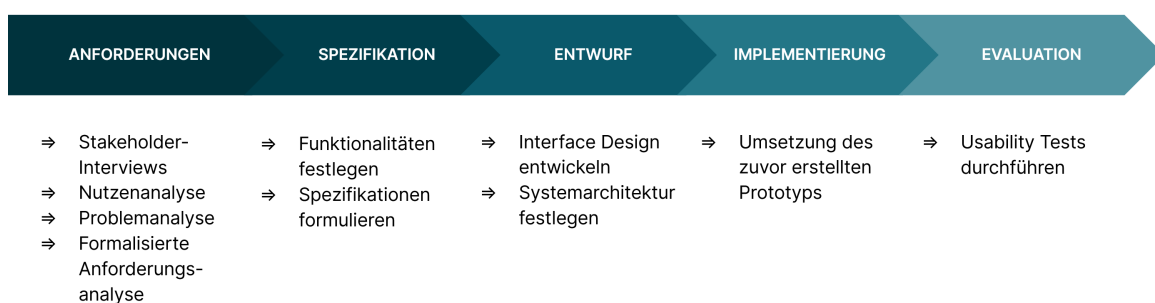
**F2 |** Was sind Anforderungen an ein System, welches die in F1 gezeigten Schwierigkeiten adressiert und reduziert?

Abschließend soll eruiert werden, ob die erste Iteration des Systems die ermittelten Schwierigkeiten mit den erarbeiteten Funktionalitäten lösen kann. Die letzte Forschungsfrage beschäftigt sich demzufolge mit der Evaluation des erarbeiteten Systems und den möglichen Stärken und Schwächen.

**F3 |** Inwieweit kann ein aus F2 resultierender Prototyp die in F1 identifizierten Schwierigkeiten reduzieren?

### 1.3 Vorgehensweise

Die Entwicklung des Systems orientiert sich am menschenzentrierten Gestaltungsprozess (DIN EN ISO 9421-210:2020-03, 2020). Der Prozess teilt sich im Rahmen dieser Arbeit in fünf aufeinanderfolgende Phasen (Abbildung 2.1), wobei die Entwurfs- und Implementierungsphasen Spielraum für ein iteratives Vorgehen lassen. Unter anderem werden in der Analyse die Aufgaben des Systems, die Benutzenden und der Kontext nach dem Entwicklungsprozess für interaktive Medien (Herczeg, 2009a) aufgeführt, um ein gebrauchstaugliches Ergebnis erzielen zu können.



**Abbildung 1.1:** Vorgehensmodell

In der ersten Phase werden die Anforderungen an das zu entwickelnde System erarbeitet und verstanden. Die Erkenntnisse dieser Phase sind in Kapitel 2 zu finden. Kapitel 2 soll unter anderem die Forschungsfragen F1 und F2 beantworten. Durch Interviews mit Stakeholdern sollen die zentralen Schwierigkeiten des Ausleihprozesses festgestellt werden. Folglich werden die Notwendigkeit der angedachten Anforderungen mithilfe der Interviews überprüft und ergänzt.

In der Spezifikationsphase werden die Anforderungen an das System weiter spezialisiert (Kapitel 3). Daher werden Funktionalitäten entsprechend den Anforderungen entwickelt und in einer priorisierten Feature-Liste festgehalten. Anschließend werden die Systemarchitektur, aufgeteilt in Frontend, Backend und den bestehenden Snipe-IT Server, erarbeitet und passende Frameworks begründet zur Entwicklung ausgewählt. Darüber hinaus wird das Interface-Design erarbeitet (Kapitel 4). Hierbei wird durch Usability Tests ein iteratives Vorgehen ermöglicht.

Die Implementierungsphase umfasst die eigentliche Umsetzung des Reservierungstools (Kapitel 5). Hierbei werden die in der Konzeptionsphase festgelegten Frameworks genutzt. Kapitel 6 präsentiert das realisierte System anhand von Dialogbeispielen.

In der abschließenden Phase werden das realisierte System mithilfe von Interviews und Umfragen evaluiert (Kapitel 7). Die Ergebnisse der Phase beantworten Forschungsfrage F3 und geben Aufschluss über die Wirksamkeit des entwickelten Systems. Im Anschluss gibt Kapitel 8 einige Perspektiven über offene Punkte und die mögliche Weiterentwicklung des Systems.

# 2

## Anforderungen

Um Benutzende, Aufgaben und Kontext des Projektes genauer zu verstehen, wurde eine Analyse nach dem menschenzentrierten Gestaltungsprozess durchgeführt (DIN EN ISO 9421-210:2020-03, 2020). Hierbei wurden relevante Anforderungen, die Einfluss auf die Entwicklung und die spätere Nutzung des Reservierungstools nehmen, identifiziert und eingeordnet. Mit der Durchführung einer Recherche, welche in den Datenquellen (Abschnitt 2.1) genauer präsentiert wird, wurden zunächst zwei Benutzergruppen (Verleihende und Ausleihende) festgehalten. Mittels einer Benutzeranalyse (Abschnitt 2.2) unter der Zuhilfenahme von durchgeführten Interviews, konnten diese weiter klassifiziert und eingegrenzt werden. Daraufhin wurden die Probleme und Herausforderungen des aktuellen Vorgehens und den unterschiedlichen Ausleihprozessen (Abschnitt 2.3) hergeleitet. Anschließend wurden die Aufgaben, die Verleihende und Ausleihende mithilfe der Anwendung bewältigen sollten, diskutiert (Abschnitt 2.4). Abschließend wurde der organisatorische und zeitlich-räumliche Kontext des Verleihs und Ausleihs am IMIS (Abschnitt 2.5) untersucht. Aufbauend auf den Resultaten der vorangestellten Untersuchungen wurden die objektiven Anforderungen an den Snipe-IT Companion nach Balzert (2009) formalisiert (Abschnitt 2.6). Diese dienen als Grundlage für das gesamte Konzept der Arbeit.

### 2.1 Datenquellen

Im Rahmen der Analyse wurden Stakeholder-Interviews durchgeführt (Unterabschnitt 2.1.1). Daraufaufgehend wurde nach vergleichbaren Projekten und Systemen recherchiert (Unterabschnitt 2.1.2).

### 2.1.1 Stakeholder-Interviews

bei denen es um die subjektive Sicht von Personen geht, die unterschiedlich ausfallen und begründet werden können – hier liegt eine Stärke qualitativer Verfahren, weil sie z.B. in Interviews dem Befragten für eigene Sinngebungen, Deutungen etc. Raum geben (vgl. Bortz Döring 2016, 184).

Zunächst wurde anhand der zu untersuchenden Gesichtspunkte ein Interviewleitfaden für ein semi-strukturiertes Interview entwickelt (Anhang A). Mithilfe dieses Leitfadens wurden die Befragten durch das Interview geführt. Die Interviews wurden aufgezeichnet und anschließend in Teilen, mithilfe eines vereinfachten Transkriptionssystems, verschriftlicht (Dresing & Pehl, 2016). Daraufhin wurde eine qualitative Inhaltsanalyse durchgeführt. Hierfür wurden die Transkripte erneut gelesen und für die Arbeit und Forschungsfragen relevante Textstellen kommentiert. Infolgedessen wurden die erarbeiteten Kommentare in ein Ordnungssystem strukturiert und unter den einzelnen Analysepunkten zusammengefasst (Dresing & Pehl, 2016).

Es wurde eine Unterteilung in Verleihende und Ausleihende von Assets vorgenommen (genauere Definition der Benutzergruppen in Abschnitt 2.2). Bei den Teilnehmenden handelt es sich um Mitarbeitende, welche am IMIS tätig sind und Studierende der Medieninformatik an der Universität zu Lübeck. In Tabelle 2.1 ist der jeweilige (Haupt-)Zuständigkeitsbereich der Verleihenden aufgeführt. Verleihende der Assets können gleichzeitig die Position eines Ausleihenden einnehmen. Die Ausleihenden umfassen außerdem Studierende, welche in Tabelle 2.2 dargestellt sind. Durch den geplanten Einsatz am IMIS wurde sich zunächst ausschließlich auf wissenschaftlichen Mitarbeiter:innen (WiMi) des IMIS und Studierende der Medieninformatik begrenzt. Hierbei wurde insbesondere der Fokus auf die Probleme am IMIS gelegt (genauere Untersuchungen in der Problemanalyse in Abschnitt 2.3). Die IDs der Teilnehmenden werden als Verweise in den folgenden Abschnitten verwendet.

**Tabelle 2.1:** Teilnehmende der Interviews, Verleihende

ID	Alter	Zuständigkeitsbereich
V1	25 - 35 J.	Keine direkte Zuständigkeit, Zugänge zu verschiedenen Laboren
V2	25 - 35 J.	Multimedialabor
V3	25 - 35 J.	VR-Labor
V4	40 - 59 J.	Administratives Personal
V5	25 - 35 J.	Innovationslabor

**Tabelle 2.2:** Teilnehmende der Interviews, Ausleihende

(die mit \* gekennzeichneten Personen wurden gemeinsam interviewt)

ID	Alter	Rolle
A1	19 - 25 J.	Bachelorstudentin, Hilfswissenschaftlerin
A2	19 - 25 J.	Bachelorstudent
A3	19 - 25 J.	Masterstudent, Hilfswissenschaftler
A4*	19 - 25 J.	Bachelorstudentin
A5*	19 - 25 J.	Bachelorstudentin
A6	19 - 25 J.	Masterstudentin

Des Weiteren wurde mithilfe des Fragebogen zur interaktionsbezogenen Technikaffinität (ATI) das technische Interesse und Verständnis der Teilnehmenden festgestellt (Tabelle 2.3) (Attig, Wessel & Franke, 2017). In beiden Gruppen konnten lediglich geringe Unterschiede innerhalb der soziodemografischen Daten festgestellt werden. Neben diesen Daten sind im Kontext der vorliegenden Arbeit insbesondere die Technikaffinität relevant.

Insgesamt konnte für beide Gruppen eine Einschätzung der Technikaffinität anhand der ATI-Skala ermittelt werden (Verleihende:  $M=5.00$ ,  $SD=0.58$ ,  $N=3$ ; Ausleihende:  $M=5.09$ ,  $SD=0.48$ ,  $N=9$ ) (Tabelle 2.3). Durch das Heranziehen zweier Vergleichsstichprobe aus Franke, Attig und Wessel (2019) ( $M=4.14$ ,  $N=300$  und  $M=4.23$ ,  $N=65$ ), lässt sich schlussfolgern, dass die Nutzendengruppen eine vergleichsweise hohe Technikaffinität aufweist.

**Tabelle 2.3:** Ergebniswerte der ATI-Skala

Benutzergruppe	Mittelwert ( $M$ )	Standardabweichung ( $SD$ )	Teilnehmende ( $N$ )
Verleihende	5.00	0.58	3
Ausleihende	5.09	0.48	9

### 2.1.2 Recherche vergleichbare Systeme

Für die Recherche wurden die digitalen Bibliotheken ACM Digital Library<sup>1</sup> und Google Scholar<sup>2</sup> genutzt. Es wurden Begriffe aus den Bereichen Managementsystem (*Assets*, *Assetmanagement*, *schedule*) und Reservation (*reservation*, *Lending*, *lend*, *borrow*) zur Suche verwendet. Die Recherche sollte dem besseren Vergleichen und Abwägen der analysierten

<sup>1</sup><https://dl.acm.org/>

<sup>2</sup><https://scholar.google.de/>

Interview-Ergebnisse dienen und vergleichbare Systeme und Projekte eruieren. Des Weiteren sollte die Recherche dazu dienen mögliche Fehlerquellen zu erörtern, um diese umgehen zu können. Durch den speziellen Einsatzort am IMIS stellte sich zeitnah heraus, dass wenige bis keine vergleichbaren Systeme gefunden werden konnten. Folglich boten die vorhandenen Vergleiche keinen hilfreichen Aufschluss für den Rahmen dieser Arbeit, daher wurde die Recherche nicht fortgeführt.

Systeme schildern: Airbnb und Otto

## 2.2 Benutzeranalyse

Um eine zielgruppengerechte Gestaltung des Snipe-IT Companion voraussetzen zu können, werden in diesem Abschnitt die Benutzergruppen des Reservierungstools eruiert und näher untersucht. Resultierend aus den Stakeholder-Interviews wurden zwei Benutzergruppen, zu denen Verleihende sowie Ausleihende eines Assets gehören, für das System erarbeitet. Die Zielgruppe beschränkt sich im Rahmen dieser Arbeit auf die Mitarbeitenden des IMIS sowie die Studierenden der Medieninformatik an der Universität zu Lübeck. Für den Snipe-IT Companion konnte eine Zielgruppe mit einer Altersspanne von 17 bis 35 Jahren festgelegt werden. Aus den Interviews (Abschnitt 2.1) konnte entnommen werden, dass beide Benutzergruppen täglich ein Smartphone, Tablet, Laptop oder Desktop PC nutzen und somit ein grundlegendes technisches Verständnis vorausgesetzt werden kann. Diese Behauptung konnte mit den Ergebnissen der ATI-Skala bestärkt werden (Tabelle 2.3).

### 2.2.1 Verleihende

Verleihende umfassen ausschließlich Mitarbeitende, sowie administratives Personal des IMIS, welche verantwortlich für die Ausgabe von Assets sind (Tabelle 2.1). Allerdings haben nicht alle Verleihende auf alle Assets den gleichen Zugriff, da dies von Forschungsgruppe zu Forschungsgruppe unterschiedlich ist. Zudem liegen in den Forschungsgruppen unterschiedliche Vorgänge vor (genauere Unterschiede zu den Vorgängen in Abschnitt 2.3). Folglich nehmen Verleihende der Assets gleichzeitig die Position der Ausleihenden ein.



### 2.2.2 Ausleihende

Bei Ausleihenden handelt es sich insbesondere um Studierende, welche keinen direkten Zugang zu den Assets haben (Tabelle 2.2). Ausleihende suchen Verleihende (Mitarbeitende) aktiv auf oder kontaktieren jene, um Informationen über ausleihbare Assets zu erhalten. Wie bereits geschildert, können durch die Forschungsgruppen auch Verleihende zu Ausleihenden werden, wobei für diese häufig ein anderer Ausleihprozess als für Studierende vorliegt (Abschnitt 2.3).

## 2.3 Problemanalyse

Um die Relevanz des Verleihs am IMIS sowie die Prozesse und die damit einhergehenden Problematiken besser nachvollziehen zu können, wurde eine Problemanalyse auf Basis des aktuellen Vorgehens durchgeführt. Hierfür wurde sich auf die in Abschnitt 2.1 aufgeführten Interview-Teilnehmenden referenziert. Der Übersichtlichkeit wegen werden zunächst Probleme, welche Verleihende und Ausleihende betreffen, thematisiert. Daraufhin werden Probleme der einzelnen Benutzergruppen näher erläutert.

### 2.3.1 Probleme: Allgemein

Eines der größten Probleme im derzeitigen Ablauf ist das Nichtvorhandensein einer öffentlichen Liste für Studierende, über die ausleihbare Assets eingesehen werden können. Auch aufseiten der Verleihenden ist keine vollständige interne Übersicht vorhanden (V1-5, A1-6). Dies führt dazu, dass aufgrund von fehlender Information wenig Assets ausgeliehen werden können (V1-5). Durch die verschiedenen Forschungsgruppen am IMIS und die damit verbundenen Labore, gibt es verschiedene Ansprechpartner:innen für die jeweiligen Assets in den Laboren. Die Verantwortlichkeiten dieser Ansprechpartner:innen ist jedoch nicht ausreichend ersichtlich, sodass es häufig zu Weiterverweisen an andere Ansprechpartner:innen kommt, wobei auch unter den WiMi nicht immer die Verantwortlichkeiten bekannt sind (V1, V3, V4, A1, A2, A3).

Studierende, welche als Hilfwissenschaftler:innen (HiWi) am IMIS angestellt sind, haben beim Ausleihen häufig einen Vertrauensvorschuss, sodass bei kurzer Ausleihe für Tätigkeiten im Gebäude Assets verliehen oder entnommen werden, ohne dies zu Vermerken. Diese Praxis gilt auch für WiMi (A1, V1, V2), wobei das Planen mit Assets dadurch erschwert wird (A3, A6).

Beim Verleihen von Assets kommt es häufig vor, dass insbesondere Studierende ohne Erfahrungen und Expertise ein Asset ausleihen wollen (V2, A3, A6). Folglich kann dies schnell zu Problemen führen. Während einige Verleihende auf die Selbstaneignung und Google verweisen, ist es anderen wichtig, den Use Case der Nutzung zu verstehen und die damit verbundenen Einstellungen der Assets sowie weiteres Zubehör zu empfehlen (V1, V2, V4, V5). Somit ist die Nutzung der Assets durch das Ungleichgewicht aufseiten der Verleihenden nicht kontrollierbar. Dies führt untereinander dazu, dass Beschädigungen, Gebrauchsspuren und Mängel von Assets nicht festgehalten werden können. Ausleihende erfahren den Zustand und die Mängel eines Assets erst zum Zeitpunkt der Abholung, was die Nutzung beeinträchtigen kann (A1, V1, V5).

### 2.3.2 Probleme: Verleihende

Eine zentrale Schwachstelle des aktuellen Vorgehens sind die uneinheitlichen Prozesse. Das Ausleihen von Assets wird von Forschungsgruppe zu Forschungsgruppe unterschiedlich gehandhabt. Zudem liegen auch innerhalb der Forschungsgruppen Unterschiede im Prozess vor (V1, V2, V3). Wie bereits in den Allgemeinen Problemen (Unterabschnitt 2.3.1) geschildert, ist es einigen Verleihenden wichtig, dass Ausleihende über die Assets Bescheid wissen und der Use Case detailliert besprochen und erläutert wird. Für andere ist die Vermittlung dieses Wissens wiederum nicht von Bedeutung (V1, V2, V3, V4) (siehe Probleme: Ausleihende).

Um ein Asset ausleihen zu können, müssen Ausleihende, je nach Zuständigkeitsbereich, entweder ein Formular oder nur auf einem beliebigen Zettel unterschreiben (V1, V2, V3, V4). Dies führt mitunter zu einer unübersichtlichen »Zettelwirtschaft« (V4, V5). Wiederum wird durch das Vertrauen am IMIS und bei kurzen Dauern nicht dokumentiert, wer oder wie lange das Gerät genutzt wird (V1, V2). Mitarbeitende, welche von anderen Forschungsgruppen ein Asset ausleihen möchten, haben häufig einen anderen Ausleihprozess als Studierende, da der Vertrauens- und Bekanntheitsgrad sich unterscheidet (V1, V2, V3). Durch die mangelnde und uneinsichtige Dokumentation des Ausleihens kann ein spontanes Planen erschwert werden (V1, V2, V3).

Da auch eine interne Übersicht der verfügbaren Assets für Verleihende fehlt, kann es durch Unwissenheit zu Doppelbeschaffung kommen (V1, V2, V3). Folglich kann es zu Anschaffungen kommen, welche mit bereits vorhandenen Assets nicht kompatibel sind (V2, V3). Eine Schwachstelle liegt im Beschaffen von Assets, ohne dass diese vermerkt werden. Dies führt zu Assets, welche in einzelnen Büros liegen oder vergessen werden, obwohl diese für laufende Studien oder Ähnliches sinnvoll sein können (V1, V3). Interviewte führten einen konkreten

Fall an, in welchem verschwunden geglaubte Assets nach zwei Jahren wieder aufgefunden wurden (V3).

Eine weitere Schwachstelle lässt sich in der Wartung von Assets feststellen. Mitarbeitende sind für die von ihnen angeschaffte Assets zuständig. Jedoch fehlen Übersichten und Erinnerungen für Wartungen und Updates der Assets, sodass es beispielsweise zur Entladung von Akkus kommen kann oder Assets nicht spontan genutzt werden können, weil keine Software-Updates durchgeführt wurden (V1, V2, V5).

### 2.3.3 Probleme: Ausleihende

Die mangelnde Sichtbarkeit der Assets nach außen erschwert das Ausleihen für Studierende. Der Austausch zu Assets findet fast ausschließlich über Übungen, Workshops oder institutsinterne Kommunikation statt (A1-5). Obwohl einige Studierende dem Ausleihen per E-Mail nicht abgeneigt sind, erschweren fehlende Informationen zur Verfügbarkeit den Prozess (A4, A5). Studierende, welche dem E-Mail-Prozess negativ gegenüberstehen, führen die langen Antwortzeiten als Hauptgrund an. Folglich sei das spontane Planen von Assets unzuverlässig (A1, A6). Spontanes Nachfragen nach Assets ist aufseiten der Studierenden weniger zu sehen. Hier wird vorher eine E-Mail an die verantwortlichen Übungsleiter:innen oder die Studiengangkoordination geschrieben, da das Risiko zu hoch sei, dass Assets nicht ausleihbar sind (A3). Stark bemängelt wird, dass die Verfügbarkeit der Assets zum gewünschten Zeitpunkt der Ausleihe nicht garantiert ist (A1, A3). Zudem kann kein schnelles Herankommen ermöglicht werden (A3). Des Weiteren wird das Nachfragen als nervig empfunden, da WiMi in ihrer Arbeit gestört werden, obwohl eine Liste der Assets ausreichen würde (A6). Ein weiterer Kritikpunkt ist, dass Ausleihende nicht über alle Geräte Kenntnis besitzen, sodass es durch fehlende Auskunft beispielsweise zu Kompatibilitätsfehlern kommen kann (A3). Dies führte dazu, dass Projekte nicht mit dem ausgeliehenen Asset umgesetzt werden konnten (A3).

### 2.4 Aufgabenanalyse

Durch die Schritte, welche Verleihende und Ausleihende im Ausleihprozess durchlaufen, konnten auf Basis der Interviews (Abschnitt 2.1) Aufgaben erarbeitet werden, welche von dem Reservierungstool übernommen oder unterstützt werden können. Die Aufgaben wurden anhand des aktuell idealen und vorgesehenen Ausleihprozesses in drei Bereiche eingeordnet. Im ersten Bereich handelt es sich um die Vorbereitungen, welche zum Ausleihen eines Assets getroffen werden müssen. Darauffolgend werden die Aufgaben der Ausgabe definiert. Der dritte Bereich umfasst die Rückgabe der Assets. Ergänzend zu den zuvor genannten Bereichen wurden Aufgaben für die Wartung der Assets dargestellt.

#### 2.4.1 Aufgaben im Bereich der Vorbereitung

Um ein Asset ausleihen zu können, müssen bestimmte Vorbereitungen getroffen werden, welche im Folgenden näher erläutert werden (Tabelle 2.4).

**Tabelle 2.4:** Aufgaben im Bereich der Vorbereitung

ID	Aufgabe
Ag-Vt-1	Nachfragen, ob Assets vorhanden
Ag-Vt-2	Abfragen der Verfügbarkeit
Ag-Vt-3	Einsehen der Verfügbarkeit
Ag-Vt-4	Reservierung der Assets
Ag-Vt-5	Beratung der Ausleihenden

**Ag-Vt-1 | Nachfragen ob Assets vorhanden**  
erläutern

**Ag-Vt-2 | Abfragen der Verfügbarkeit**

Um ein Asset ausleihen zu können, muss eine Anfrage an die Verantwortlichen gesendet werden. Dies geschieht meist per E-Mail. Ausleihende fragen nach konkreten Assets, dessen Existenz sie bspw. durch die institutsinterne Kommunikation erfahren haben. Wie bereits in der Problemanalyse geschildert (Probleme: Allgemein), gibt es keine Übersicht über ausleihbare Assets. Dies zeigt die Dringlichkeit des Snipe-IT Companion für eine bessere Vorbereitung.

**Ag-Vt-3 | Einsehen der Verfügbarkeit**

Um ein Asset ausleihen zu können muss das gewünschte Asset ausleihbar sein. Verleihende überprüfen, ob das angefragte Asset im Schrank vorhanden ist. Hierbei ist zu berücksichtigen,

dass keine langfristige Planung gewährleistet werden kann. Dies kann durch den Snipe-IT Companion mittels eines Kalenders und einer Reservierungsfunktion ermöglicht werden.

#### Ag-Vt-4 | Reservierung der Assets

Wie in *Ag-Vt-3* bereits geschildert, kann eine langfristige Planung in die Zukunft nicht gewährleistet werden. Liegt eine Anfrage in der Zukunft vor, wird diese mittels eines Klebezettels am gewünschten Asset vermerkt. Die Verfügbarkeit des Assets zum reservierten Zeitraum hängt somit davon ab, ob die Notiz von anderen WiMi berücksichtigt wird.

#### Ag-Vt-5 | Beratung der Ausleihenden

Die zuvor erhaltene Anfrage aus *Ag-Vt-2* wird von Verleihenden bearbeitet, wobei in einigen Fällen der Use Case des Ausleihens erfragt wird, um den Ausleihenden Empfehlungen zu geben. Um den Ausleihprozess für Verleihende zu erleichtern, kann der vorangestellte Use Case mittels eines Dialogs im Snipe-IT Companion ermöglicht werden. So können aufbauend auf dem Dialog, direkt Vorschläge an Ausleihende gegeben werden.

### 2.4.2 Aufgaben im Bereich der Ausgabe

Im nächsten Abschnitt werden alle zentralen Aufgaben aufgeführt, welche für die Übergabe von Assets relevant sind (Tabelle 2.5).

**Tabelle 2.5:** Aufgaben im Bereich der Ausgabe

ID	Aufgabe
Ag-Au-1	Abholung der Assets
Ag-Au-2	Erläuterung der Assetnutzung
Ag-Au-3	Unterschreiben des Formulars

#### Ag-Au-1 | Abholung der Assets

Ausleihende holen die Assets zum besprochenen Zeitpunkt in den Laboren des IMIS ab. Verleihende schaffen hierfür Zugriff zum Labor oder Schrank.

#### Ag-Au-2 | Erläuterung der Assetnutzung

Hierbei wissen Ausleihende vorher häufig nicht genau, um was für ein Gerät es sich handelt, sodass es zu Kompatibilitätsfehlern kommen kann und das Asset für den ursprünglichen Gebrauch nicht nutzbar ist. Im besten Fall ist eine Übersicht mit Informationen wie Name, Seriennummer, Stückzahl und die dazugehörige Anleitung bereits vor dem Ausleihprozesse verfügbar, sodass Ausleihende sich im Vorhinein selbständig besser informieren können.

### Ag-Au-3 | Unterschreiben des Formulars

Im Idealfall werden die ausgeliehenen Assets in einem Formular dokumentiert, welches von den Ausleihenden unterzeichnet wird. Das Formular wird bis zur Rückgabe aufbewahrt. Durch die unterschiedlichen Vorgänge ist insbesondere diese Aufgabe nicht einheitlich.

### 2.4.3 Aufgaben im Bereich der Rückgabe

Die nachfolgenden Aufgaben umfassen die Rückgabe der ausgeliehenen Assets (Tabelle 2.6).

**Tabelle 2.6:** Aufgaben im Bereich der Rückgabe

ID	Aufgabe
Ag-Rg-1	Rückgabe der Assets
Ag-Rg-2	Überprüfung der Assets

#### Ag-Rg-1 | Rückgabe der Assets

Für die Rückgabe der ausgeliehenen Assets wird auf dem Formular dokumentiert, wann und was zurückgegeben wurde. Die Rückgabe wird meist während der Abholung oder per E-Mail besprochen. Sollte die Ausleihe ohne Formular erfolgt sein, wird der Klebezettel mit der Unterschrift entsorgt oder die Assets ohne weitere Dokumentation in die Schränke und Räume zurückgebracht.

#### Ag-Rg-2 | Überprüfung der Assets

Für einige Verleihende fällt das Überprüfen der Assets nach einer Rückgabe an. Die Überprüfung umfasst bspw. das Formatieren von SD-Karten und Laden von Akkus. Außerdem sollten Einstellungen an den Assets zurückgesetzt werden. Idealerweise würde diese Aufgabe von Ausleihenden übernommen werden, jedoch fehlt auch hier die Aufklärung, welche vom Snipe-IT Companion übernommen werden kann.

### 2.4.4 Aufgaben im Bereich der Wartung

Im Folgenden werden Aufgaben, welche für Verleihende auf administrativer Ebene von Bedeutung sind näher erläutert (Tabelle 2.7).

**Tabelle 2.7:** Aufgaben im Bereich der Wartung

ID	Aufgabe
Ag-Wt-1	Pflege von Assets
Ag-Wt-2	Pflege von Neuanschaffung

#### Ag-Wt-1 | Pflege von Assets

Assets, welche längere Zeit nicht genutzt werden, müssen von Verleihenden gewartet werden. Jedoch wird diese Aufgabe in manchen Fällen vergessen.

#### Ag-Wt-2 | Pflege von Assets

Bei Neuanschaffungen sollten WiMi über diese informiert werden. Innerhalb von Forschungsgruppen gerät dies häufig in Vergessenheit, da keine Übersicht vorhanden ist.

zuordnung  
der Aufga-  
ben zu den  
relevanten  
Nutzungs-  
phasen

## 2.5 Kontextanalyse

Für die Ermittlung der Nutzungsumgebung, in der das System verwendet werden soll, wurde eine Kontextanalyse, basierend auf den vorangehenden Analysen und Interviews durchgeführt. Zunächst wurde der organisatorische Kontext des Systems festgehalten. Anschließend wurde der zeitlich-räumliche Kontext eruiert (Herczeg, 2018).

### 2.5.1 Organisatorischer Kontext

Unter Berücksichtigung von sozialen Strukturen kann die Qualität des Systems maßgeblich positiv beeinflusst werden (Herczeg, 2018).

Innerhalb des Universitäts-Kontextes gibt es aus formeller Sicht eine überwiegend flache Hierarchie zwischen Studierenden und Mitarbeitenden, wobei zwischen Hilfwissenschaftlern:innen, wissenschaftlichen Mitarbeitenden und Professor:innen unterschieden werden kann. Diese Gruppen weisen teilweise verschiedene Zugriffe auf Labor und Schränke auf, welche die Assets beinhalten.

Um die Aufgaben im Bereich der Wartung berücksichtigen zu können, sollten Verleihende, zum Eintragen neuer Assets einen administrativen Zugang zum System erhalten (*Ag-Wt-2*). Des Weiteren sollte ein Überblick über Updates oder Ähnliches gegeben werden können (*Ag-Wt-1*).

### 2.5.2 Zeitlich-Räumlicher Kontext

Der zeitlich-räumliche Kontext sollte sowohl aus Sicht der Verleihenden als auch aus Sicht der Ausleihenden analysiert werden, da das System einen einheitlichen Ausleihprozess schaffen soll.

#### Verleihende

Mitarbeitende halten sich entweder in Präsenz an der Universität oder im Homeoffice auf. Daher werden bei der Analyse des zeitlich-räumlichen Kontextes beide Fälle betrachtet.

Befinden sich Mitarbeitende im Büro, arbeiten diese an einem Desktop-Arbeitsplatz. Der Computer ist dabei die meiste Zeit eingeschaltet, wodurch ein System in Form einer Web-App sinnvoll wäre (V1,V5). Wenn Mitarbeitende das Büro verlassen, um ein Asset zu Verleihen, können sich WiMi in verschiedenen Laboren befinden. Da der Ort der Nutzung unter anderem durch Homeoffice variiert, liegt ein mobiler Nutzungskontext vor, welcher beispielsweise durch die Nutzung einer Web-App auf dem Smartphone ermöglicht werden kann (V1, V2, V3, V5). Die Bedienung der Anwendung sollte niedrigschwellig sein, da Mitarbeitende häufig nicht viel Zeit für die Bedienung haben oder investieren möchten (V1, V2, V3). Das System sollte einen pragmatischen Zweck erfüllen und kein zu großes Konzept umfassen, sodass womöglich neue Anläufe dazu kommen und die Arbeit zweckmäßig erhöht statt reduziert wird (V2).



### Ausleihende

Studierende arbeiten flexibel zum Beispiel von Zuhause aus oder in der Bibliothek. Es wird jedoch selten am IMIS direkt gearbeitet. Demzufolge sollte das Buchen spontan, jederzeit und ortsunabhängig möglich sein. Folglich liegt ein mobiler Nutzungskontext vor, welcher beispielsweise durch die Nutzung einer Web-App auf dem Smartphone ermöglicht werden kann (A1, A3, A6).

## 2.6 Formalisierte Anforderungen

Im Folgenden werden systematisch formalisierte Anforderungen präsentiert, welche die Ergebnisse der Analysen abschließend zusammenfassen. Es werden zunächst die Visionen und Ziele (Unterabschnitt 2.6.1) definiert, des Weiteren werden die Rahmenbedingungen (Unterabschnitt 2.6.2) und der Kontext des Systems (Unterabschnitt 2.6.3) dargestellt. Darauf aufbauend wird eine funktionale Anforderung erstellt (Unterabschnitt 2.6.4). Abschließend werden die Qualitätsanforderungen formuliert (Unterabschnitt 2.6.5).

### 2.6.1 Vision und Ziele

Zunächst werden die Visionen und Ziele des Systems konkretisiert, an denen sich die Anforderungen auf Zielgerichtetheit überprüfen lassen (Balzert, 2009). Diese setzen sich aus der Analyse der Benutzenden sowie Aufgaben und des Kontextes zusammen. Zunächst werden die Visionen für die Zukunft realitätsnah festgelegt.

---

/V10/	Der Ausleihprozess von Assets am IMIS verläuft einheitlich.
/V20/	Assets des IMIS sind allen Studierenden und Mitarbeitenden bekannt und werden von beiden Gruppen genutzt.
/V30/	Der Snipe-IT Companion unterstützt Ausleihende effizient mit individuellen und anwendungsspezifischen Assetvorschlägen.
/V40/	Die Planung und Kommunikation zwischen Verleihenden und Ausleihenden verläuft reibungslos.
/V50/	Verleihende fühlen sich durch den Snipe-IT Companion unterstützt.

---

## 2 Anforderungen

---

Basierend auf diese Visionen lassen sich die Ziele formulieren, welche die Visionen operationalisieren. Diese folgen dabei den standardisierten Regeln zur Formulierung von Zielen (Pohl et al., 2008).

- 
- |       |   |
|-------|---|
| /Z10/ | Verleihende und Ausleihende sollen jederzeit in der Lage sein, ein gebrauchstaugliches, niedrighschwelliges Interface zum Ausleihen von Assets verwenden.                           |
| /Z20/ | Ausleihende sollen jederzeit standortunabhängig in der Lage sein, die Verfügbarkeit von Assets einsehen zu können und diese zu buchen.  |
| /Z30/ | Ausleihende eines Assets sollen jederzeit zielgerichtete und aktuelle Informationen zum Asset erhalten.   |
| /Z40/ | Ausleihende sollen jederzeit in der Lage sein, sich über Assets zu informieren, um initiale Nutzungsbarrieren zu überwinden und auf die Nutzung des Assets vorzubereiten.           |
| /Z50/ | Verleihende sollen jederzeit in der Lage sein, vom System gesammelte Daten übersichtlich und strukturiert einzusehen.   |
| /Z60/ | Verleihende sollen jederzeit standortunabhängig in der Lage sein, alle vorhandenen Assets am IMIS einzusehen, sodass es zu keinen unbeabsichtigten Doppelbeschaffungen kommen kann. |
| /Z70/ | Das System soll Informationen zugänglich präsentieren.  |
- 

### 2.6.2 Rahmenbedingungen

Die Rahmenbedingungen legen organisatorische und technische Restriktionen für das System oder den Entwicklungsprozess fest (Balzert, 2009). Die Bedingungen wurden aus der Benutzer- und Kontextanalyse abgeleitet.

- 
- |       |   |
|-------|---|
| /R10/ | Das System ist eine informative Web-Anwendung (Abschnitt 2.5 Kontextanalyse).               |
| /R20/ | Die Zielgruppe sind Mitarbeitende des IMIS und Studierende (Abschnitt 2.2 Benutzeranalyse). |

- /R30/ Die Zielgruppe teilt sich in zwei Nutzergruppen: die Verleihenden und Ausleihende von Assets. Die Definitionen der Nutzergruppen sind in Abschnitt 2.2 Benutzeranalyse zu finden.
- /R40/ Das System wird von Verleihenden in einem Arbeitsplatzsystemkontext und mobilen Kontext genutzt. Von Ausleihenden vorwiegend nur im mobilen Kontext (Abschnitt 2.5 Kontextanalyse).
- /R50/ Das System soll sich vorwiegend im Dauerbetrieb befinden (Unterabschnitt 2.5.2 Zeitlich-Räumlicher Kontext).
- /R60/ Das System muss unbeaufsichtigt zuverlässig lauffähig sein.
- /R70/ Die eingesetzte Software ist clientseitig ein Webbrowser. Die marktführenden Webbrowser müssen unterstützt werden: Chrome, Firefox, Safari (*Browser stats: Browser market share*, o. J.).
- 

### 2.6.3 Kontext und Überblick

Ein System ist in einer technischen Umgebung eingebettet (Balzert, 2009). Es wurde im folgenden Bezug auf das aktuelle Vorgehen mithilfe des Abschnitt 2.3 Problemanalyse geschlossen.

- 
- /K10/ Zur Anmeldung und Abruf von Informationen existiert eine Schnittstelle zum IDM LDAP.
- /K20/ Es existieren von Forschungsgruppe zu Forschungsgruppe unterschiedliche Ausleihprozesse.
- /K30/ Es existieren Formulare, mit denen das Verleihen dokumentiert wird.
- /K40/ Im Rahmen eines Pilotprojekts existiert das Asset-Management-System Snipe-IT.
-

### 2.6.4 Funktionale Anforderungen

Im Folgenden werden die Kernfunktionalitäten des Systems aufgeführt (Balzert, 2009). Diese ergeben sich aus allen Teilanalysen und den festgelegten Zielen. Um die Anforderungen mit einer eindeutigen Semantik zu formulieren, wurde eine Anforderungsschablone (Abbildung 2.1) verwendet, um natürlichsprachliche Anforderungen zu definieren (Balzert, 2009).

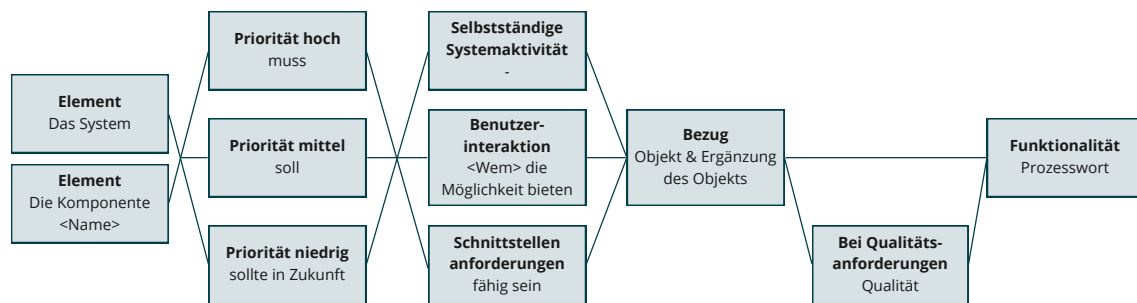


Abbildung 2.1: Anforderungsschablone (Balzert, 2009)

- 
- /F10/ Das System *muss* Verleihenden und Ausleihenden die Möglichkeit bieten, alle Assets jederzeit mittels einer Übersicht einsehen zu können (*Ag-Vt-1*, /Z10/).
- /F20/ Das System *muss* Verleihenden und Ausleihenden die Möglichkeit bieten, die Verfügbarkeit eines Assets einsehen zu können (*Ag-Vt-2*, /Z20/).
- /F30/ Das System *muss* Verleihenden und Ausleihenden die Möglichkeit bieten, nach Assets zu Filtern und zu Suchen.
- /F40/ Das System *muss* Verleihenden und Ausleihenden relevante Informationen zu den Assets anzeigen (Bild, Name, Beschreibung und Seriennummer) (*Ag-Vt-4*, *Ag-Au-2*, /Z20/, /Z40/).
- /F50/ Das System *muss* Ausleihenden relevante Informationen zu Ansprechpartner:innen anzeigen (*Ag-Vt-4*).
- /F60/ Das System *muss* Verleihenden und Ausleihenden die Möglichkeit bieten, Reservierungen von Assets vornehmen zu können (*Ag-Vt-3*).
- /F70/ Das System *muss* Verleihenden und Ausleihenden die Möglichkeit bieten, die verfügbaren Zeitslots der Assets einsehen zu können (*Ag-Vt-2*).
- /F80/ Das System *muss* Ausleihende daran erinnern, die ausgeliehenen Assets abzuholen, zurückzubringen oder zu verlängern sind (*Ag-Au-1*, *Ag-Rg-1*).
- /F90/ Das System *soll* Verleihenden und Ausleihenden die Möglichkeit bieten, sich mit dem vorhanden IDM (LDAP) Account einzuloggen.
- /F100/ Das System *soll* Ausleihenden die Möglichkeit bieten, mittels einer Nutzen-Suche individuelle und personalisierte Asset-Vorschläge zu erhalten (*Ag-Vt-3*, *Ag-Au-2*).
- /F110/ Das System *soll* Verleihende automatisch kontaktieren, wenn ein Asset reserviert wurde (*Ag-Vt-1*).
- /F120/ Das System *soll* Verleihende automatisch erinnern, wenn der Zugriff zu einem Asset benötigt wird (*Ag-Au-1*).
- /F130/ Das System *soll* Verleihenden und Ausleihenden die Möglichkeit bieten, Mängel und Schäden am Asset zu Kennzeichen.
- /F140/ Das System *sollte in Zukunft* Verleihenden die Möglichkeit geben administrative Aufgaben zu erledigen und an Wartungen zu erinnern (*Ag-Rg-2*).

- /F150/ Das System *sollte in Zukunft* Verleihenden und Ausleihenden die Möglichkeit bieten, Assets mithilfe eines QR-Scans in den Reservierung-Checkout zu packen.
- /F160/ Das System *sollte in Zukunft* Verleihenden und Ausleihenden die Möglichkeit bieten, Kommentare und Erfahrungsberichte unter Assets zu schreiben.
- /F170/ Das System *sollte in Zukunft* Kommunikation mit Verleihenden ermöglichen, sodass keine extra Instanz benötigt wird.
- 

### 2.6.5 Qualitätsanforderungen

Im letzten Schritt werden die nicht-funktionalen Anforderungen festgelegt, welche die qualitativen oder quantitativen Eigenschaften eines Systems darstellen (Balzert, 2009). Auch hier wird, falls möglich, die Anforderungsschablone aus Abbildung 2.1 verwendet.

- 
- /Q10/ Das System *muss* den Grundsätzen der DIN EN ISO 9241-110:2019-09 (Ergonomie der Mensch-System-Interaktion - Teil 110: Interaktionsprinzipien) folgen (*DIN EN ISO 9241-110*, 2019).
- /Q20/ Das System *muss* die definierten Nutzungsklassen aus Abschnitt 2.2 unterscheiden und die dazugehörigen Zugriffsrechte sicherstellen.
- /Q30/ Das System *muss* zuverlässig und ohne Störung im Dauerbetrieb laufen (Zeitlich-Räumlicher Kontext).
- /Q40/ Das System *soll* modular strukturiert sein, damit Inhalte und Funktionalitäten effizient eingebunden werden können und das System einfach erweiterbar ist.
- /Q50/ Das System *soll* beim Zugriff über das Internet eine gesicherte Übertragung (bspw. HyperText Transfer Protocol Secure (HTTPS)) ermöglichen.
- /Q60/ Das System *soll* alle Benutzerinteraktionen in unter fünf Sekunden ausführen.
-

# 3

## Konzeption

In diesem Kapitel werden die zuvor erarbeiteten Erkenntnisse und Anforderungen genutzt, um konkrete Funktionalitäten zu definieren. Anschließend wird auf die Systemarchitektur und Wahl der Frameworks eingegangen.

### 3.1 Funktionalität

Im Folgenden werden die für das System relevanten Funktionalitäten aufgeführt. Jene beschreiben, wie das System die erarbeiteten Anforderungen erfüllt. Zur besseren Lesbarkeit wurden die Funktionalitäten nach Benutzergruppen aufgeteilt.

#### 3.1.1 Funktionalitäten für Ver- und Ausleihende

Zu Beginn werden Funktionalitäten, welche für Verleihende und Ausleihende von Bedeutung sind, näher erläutert (Tabelle 3.1).

##### Ft-VA-1 | Authentifizierung über LDAP

Mithilfe der Verknüpfung zum LDAP-System der Universität zu Lübeck, können sich Nutzende mit einem bereits existierenden IDM Account einloggen. Folglich muss kein neues Konto erstellt werden. Außerdem kann überprüft werden, ob es sich bei der Anmeldung um Studierenden oder WiMi handelt, um entsprechende Systemrechte zu vergeben. Des Weiteren verhindert die Nutzung des LDAP-Systems das Eindringen unbefugter Personen.

##### Ft-VA-2 | Übersicht über ausleihbare Assets

Die Übersicht, der am IMIS vorhandenen Assets wurde mittels Kategorien umgesetzt. Dafür gibt es eine Übersicht, bei der alle Assets eingesehen werden können. Die einzelnen

deutlicheren  
Bezug zu  
den Anfor-  
derungen  
herstellen

**Tabelle 3.1:** Funktionalitäten für VA (Ver- und Ausleihende)

ID	Titel	Anforderungen
Ft-VA-1	Authentifizierung über LDAP	/F70/ /F80/
Ft-VA-2	Übersicht über ausleihbaren Assets	/V20/ /Z20/ /F50/ /K10/ /F10/ /F30/
Ft-VA-3	Verfügbarkeit von Assets	/V20/ /Z20/ /F50/ /K10/ /F10/ /F30/
Ft-VA-4	Zuständigkeitsbereich	/F50/
Ft-VA-5	Benachrichtigungen & Erinnerungen	/F100/ /F110/ /F120/
Ft-VA-6	Material-Suche	/V20/ /Z20/ /K10/ /F10/ /F30/
Ft-VA-7	Filtern und Sortieren	/V30/ /F30/ /F70/
Ft-VA-8	Detailansicht	/V50/ /Z30/ /F40/ /F50/

Kategorien beinhalten Unterkategorien. In der Übersicht werden Informationen wie Name, Seriennummer, Marke und Status eines Assets angezeigt.

#### Ft-VA-3 | Verfügbarkeit von Assets

Der Status eines Asset (verfügbar und herausgegeben) muss klar ersichtlich sein. Dies geschieht mittels Color-Coding.

#### Ft-VA-4 | Zuständigkeitsbereich

Um für Ausleihende Kontaktinformationen anzeigen zu können (*Ft-A-6*), müssen die Zuständigkeitsbereiche eingetragen werden können. Außerdem sollten alternative Ansprechpartner:innen kenntlich gemacht werden, um Abholtermine aufgrund von Krankheit oder Homeoffice nicht verschieben zu müssen. Für Rückfragen zu einem Asset sind Kontaktinformation zu Ansprechpartner:innen (Verleihende) sowie E-Mail-Adressen hinterlegt.

#### Ft-VA-5 | Benachrichtigungen & Erinnerungen

Verleihende werden nach dem Ausleihen eines von ihnen verantwortlichen Assets benachrichtigt. Die Benachrichtigung umfasst, wer das Asset wann reserviert hat und wann die Abholung für das Asset stattfinden soll. Außerdem wird es eine direkte Weiterleitungsmöglichkeit geben, sollten Verleihende verhindert sein. Nachdem die Reservierung eines Assets stattgefunden hat, erhalten Ausleihende eine Zusammenfassung über die Ausleihdaten und einen Hinweis, wann die Abholung stattfindet. Außerdem werden Kontaktinformation des Verleihenden angezeigt (Name, E-Mail und Lagerort des Assets). Zusätzlich erhalten Verleihende und Ausleihende eine Erinnerung, sobald die Abholung oder Rückgabe eines Assets ansteht.

Design-Entscheidungen begründen und nicht nur darstellen. D.h. hier vielleicht einen Satz dazu, warum du Color Coding als geeignete Variante dafür identifiziert hast



### Ft-VA-6 | Material-Suche

Die Material-Suche umfasst eine Einteilung der Assets nach Kategorie, sowie die gezielte Suche nach der Verfügbarkeit der Assets (verfügbar und herausgegeben). Das gezielte Suchen nach Verfügbarkeit wird durch die Aufforderung, den Ausleihzeitraum anzugeben, ermöglicht. Daraufhin gibt es die Möglichkeit gewünschte Materialien in einem Suchfeld einzugeben oder über die Kategorien nach dem Asset zu suchen. Das Suchfeld unterstützt bereits während der Eingabe mithilfe von Assetvorschlägen. Die Vorschläge umfassen Assets oder Kategorien.

### Ft-VA-7 | Filtern und Sortieren

Um das Finden der Assets leichter zu gestalten, sollen Nutzende stets nach Kategorie, Nutzen und Verfügbarkeit filtern können. Außerdem ist das Sortieren des Titels von A-Z oder Z-A möglich. Um bei der Anzeige einen Überblick über die Menge der Assets behalten zu können, wird die Anzahl der Assets in der ausgewählten Kategorie angezeigt.

### Ft-VA-8 | Detailansicht

In der Detailansicht werden die Assets und deren Eigenschaften dargestellt. Hierbei werden Informationen wie Name, Seriennummer, Artikelbeschreibung, technische Details und Kontaktinformation der Verleihenden dargestellt. Des Weiteren kann der Ausleihzeitraum eingestellt und die Zeiträume der Verfügbarkeit eines Assets eingesehen werden. In Form eines Button wird sichtbar, dass das Asset zur Ausleihe hinzugefügt werden kann. Das Hinzufügen wird durch eine Pop-Up-Mitteilung bestätigt.

## 3.1.2 Funktionalitäten für Verleihende

Im Folgenden werden Funktionalitäten, welche für Verleihende von Bedeutung sind näher erläutert (Tabelle 3.2).

**Tabelle 3.2:** Funktionalitäten für (V)erleihenden

ID	Titel	Anforderungen
Ft-V-1	Dashboard	???
Ft-V-2	Bearbeiten des Assetstatus	/F150/
Ft-B-4	Kalenderansicht für Verleihende	/V50/ /Z30/ /F40/ /F50/
Ft-V-3	Pflege von Assets	/F130/
Ft-V-4	Pflege der Datenbank	/F140/

Auch hier wieder die Frage - warum hast du diese Umsetzung gewählt?

deutlicheren Bezug zu den Anforderungen herstellen

#### Ft-V-1 | Dashboard

Die Dashboard-Ansicht umfasst eine Aufgabenliste des ausgewählten Tages. Die Aufgaben umfassen Abholungs- oder Rückgabetermine sowie Wartungserinnerungen. Über eine Wochenliste kann zwischen den Tagen gewechselt werden. Außerdem werden die aktuellsten Benachrichtigungen angezeigt.

#### Ft-V-2 | Bearbeiten des Assetstatus

Assets, welche an Ausleihende übergeben wurden, müssen zunächst manuell in ihrem Status bestätigt werden. Hierfür ist eine simple Button-Funktion vorgesehen, sodass der Status des Assets schnell aktuell gehalten werden kann. Außerdem soll das System, nachdem eine Abholung oder Rückgabe stattgefunden hat, eine Benachrichtigung an Verleihende senden, sollte der Status nicht bereits bestätigt worden sein.

#### Ft-B-4 | Kalenderansicht für Verleihende

Die Kalenderansicht für Verleihende beinhaltet eine Übersicht über alle Assets, welche in diesem Moment verliehen sind. Mithilfe einer Monatsübersicht werden Termine zur Asset-Abholung, Rückgabe oder zur Wartung angezeigt. In einer Wochenansicht auf dem Dashboard gibt es eine detailreichere Ansicht eines jeweiligen Tages.

#### Ft-V-3 | Pflege von Assets

Mit Hilfe der Zuständigkeitsbereich (*Ft-V-2*) kann die Pflege der Assets besser kontrolliert werden. Außerdem können durch Erinnerungen und Checkliste Wartungen weniger leicht vergessen werden und besser aufgeteilt werden.

#### Ft-V-4 | Pflege der Datenbank

Diese Funktionalität wird bereits durch Snipe-IT bereitgestellt. Jedoch sollten Anleitungen erstellt werden, um die Pflege zu erleichtern.

### 3.1.3 Funktionalitäten für Ausleihende

Im Folgenden werden Funktionalitäten, welche für Ausleihende von Bedeutung sind, näher erläutert (Tabelle 3.3).

#### Ft-A-1 | Startübersicht

Die Startseite soll Nutzenden helfen, einen Überblick zu erlangen. Für Erstnutzende sind Hinweise für die Material-Suche in Form von Buttons gegeben. Für Nutzende, die bereits Assets ausgeliehen haben, wird eine Übersicht über laufende, kommende und vergangene Reservierungen angezeigt. Wichtige Informationen, wie der Zeitraum, werden direkt auf einen Blick ersichtlich.

baue ich  
eigentlich  
nicht mit ein

Das hier  
verstehe ich  
noch nicht  
so ganz.  
Hier wären  
glaube ich  
zwei drei  
Sätze mehr  
ganz hilf-  
reich

**Tabelle 3.3:** Funktionalitäten für (A)usleihende

ID	Titel	Anforderungen
Ft-A-1	Startübersicht	/F60/
Ft-A-2	Reservierungs-Checkout	/F60/ /F150/
Ft-A-3	Rückgabe-Checkliste	???
Ft-B-4	Kalenderansicht für Ausleihende	/V50/ /Z30/ /F40/ /F50/

**Ft-A-2 | Reservierungs-Checkout**

Mithilfe des Reservierungs-Checkouts können alle ausgewählten Assets überblickt werden. Außerdem werden alle Ausleihdaten, wie Zeitraum der Ausleihe, Abholung und Rückgabe aufgeführt. Des Weiteren gibt es die Möglichkeit alle Ausleihdaten bearbeiten zu können, sollte ein Datum oder eine Uhrzeit unpassend sein. Abschließend werden die Regeln und Sicherheitshinweise aufgeführt. Nach Bestätigung einer Erklärung zur Reservierung, gilt die Reservierung des Assets als abgeschlossen.

**Ft-A-3 | Rückgabe-Checkliste**

Bevor die ausgeliehenen Assets an Verleihende zurückgeben werden, wird eine Checkliste für das jeweilige Asset angezeigt. Dort werden einige Hinweise angezeigt, wie zum Beispiel SD-Karte geleert, Assets auf Ursprungseinstellungen zurückgestellt oder Akkus geladen. Diese Funktionalität soll insbesondere dafür sorgen, dass nachfolgende Ausleihende die Assets direkt nutzen können.

**Ft-B-4 | Kalenderansicht für Ausleihende**

Die Kalenderansicht für Ausleihende beinhaltet eine Übersicht, über alle Assets, welche gerade ausgeliehen sind. Mithilfe einer Monatsübersicht werden Termin zur Asset-Abholung oder Rückgabe angezeigt.

Mehr arbeit.. eher raus nehmen?

## 3.2 Systemarchitektur

Die Systemarchitektur gibt eine Übersicht über die technische Umsetzung des Systems und bildet die Basis der Realisierung von Software-Systemen (Dumke & Lehner, 2000). Diese besteht im Wesentlichen aus den folgenden drei Komponenten (??): dem *Snipe-IT Server*, dem Reservierungsinterface und dem Front-End.

Aufbauend auf den Anforderungen und der am IMIS bereits eingesetzten Asset Managementsoftware *Snipe-IT* werden im Folgenden die gewählten Frameworks erläutert. Um ein besseres Verständnis für die Architektur des entwickelten Software-Systemes voraussetzen zu

können, wird nach dem C4-Modell zur Visualisierung von Softwarearchitektur der Aufbau der Architektur mit den Teilkomponenten detaillierter dargestellt (Brown, 2013).

### 3.2.1 Frameworkwahl

Die Frameworkwahl nimmt durch die unterschiedlichen Arbeitsweisen und Funktionen der Frameworks enormen Einfluss auf den Entwurf eines Systems und wird daher im Folgenden näher erläutert.

Die Grundlage der Auswahl der im Rahmen dieser Arbeit eingesetzten Frameworks bilden die eingangs beschriebenen Anforderungen (Abschnitt 2.6). Dem System wird vorausgesetzt, dass es sich um eine Web-Anwendung mit Fokus auf den Einsatz im mobilen Kontext (/R10//R40/) handelt. Für Nutzende ist es wichtig, dass das System dauerhaft erreichbar ist (/R50/). Aus funktionaler Sicht müssen die Frameworks eine Unterstützung für progressive Web-Applikationen bieten. Folglich ist auch eine Unterstützung für HTTPS notwendig (/Q50/). Außerdem sollte es einfache Möglichkeiten zur Verknüpfung von LDAP bieten (/K10/ /F90/).

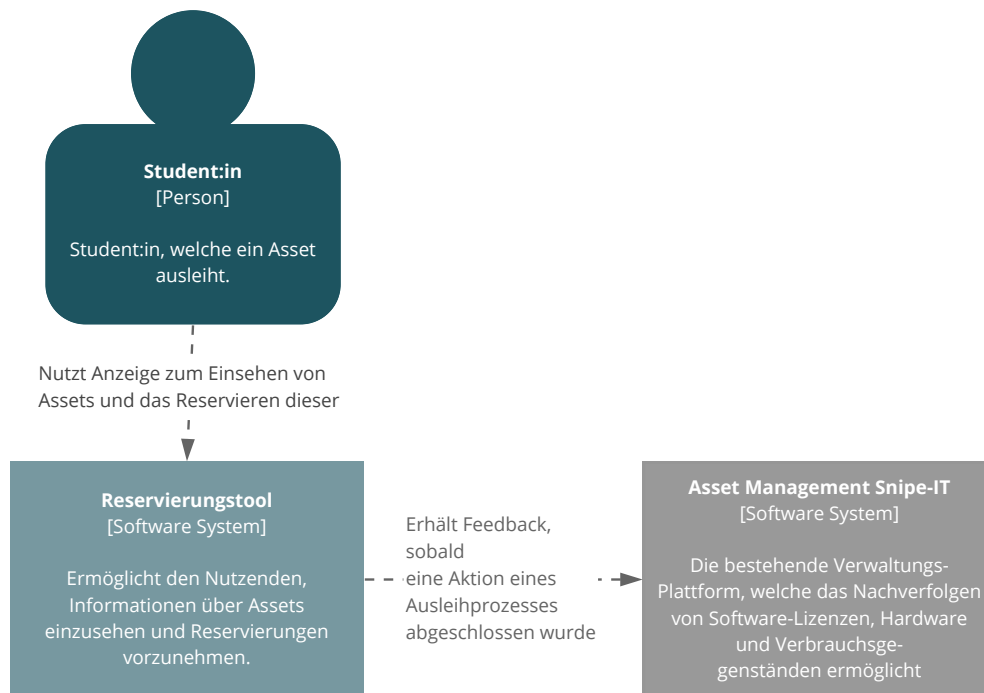
### 3.2.2 C4-Modell zur Visualisierung von Softwarearchitektur

Das Modell unterteilt die Architektur in vier Abstraktionsebenen (Level): **Context**, **Container**, **Components** und **Code**. Wobei im folgenden ausschließlich auf die ersten drei Ebenen eingegangen wird. Des Weiteren werden die Bestandteile des Systems in vier Kategorien gegliedert:

- **Person:** Stellt Nutzende eines Software-Systems dar.
- **Software-System:** Stellt die höchste Abstraktionsebene der Software dar.
- **Container:** Stellt einzeln ausführbare Teilkomponenten des Software Systems dar.
- **Component:** Stellt z.B. die Datenbank oder Anwendung dar.

#### Level 1: Context

Das erste Level der Architekturvisualisierung, nach dem C4-Modell, stellt die entwickelte Software mithilfe eines Systemkontextdiagramms dar. Abbildung 3.1 zeigt die zwei Komponenten des im Rahmen der Arbeit entwickelten Systems im Zusammenhang mit der Asset Managementsoftware *Snipe-IT*.



**Abbildung 3.1:** Architektur Level 1 nach Brown (2013)

Der Kontext umfasst zwei Software-Systeme: das Reservierungstool und die Asset Managementsoftware Snipe-IT. Das Reservierungstool bildet die Oberfläche für Ausleihende und ermöglicht es, die Assets einzusehen, zu suchen und zu buchen. Außerdem laufen über die Web-Oberfläche alle administrativen Aufgaben für Verleihende, wie das Aktualisieren eines Asset-Status. Hierzu nutzt die Web-App das Reservierungsinterface, um auf die *Snipe-IT* Funktionalitäten zuzugreifen.

Referenzierungen hinzufügen

Die Basis für das in dieser Arbeit umgesetzte System schafft die Asset Managementsoftware *Snipe-IT* (Snipe-IT, o. J.), welche bereits am IMIS eingesetzt wird. *Snipe-IT* ist eine kostenlose, quelloffene IT-Asset-Verwaltungs-Plattform, welche das Nachverfolgen von Software-Lizenzen, Hardware und Verbrauchsgegenständen ermöglicht. Genannte Assets können über ein Dashboard hinzugefügt, verwaltet und gelöscht werden. Über Labels können Assets für die Übersichtlichkeit in verschiedene Kategorien eingeordnet werden, während Tags ein Asset eindeutig identifizieren (z. B. Seriennummer). Zudem ermöglicht das „Checkin/Checkout“-System die Nachverfolgung aller Assets, falls diese zum Beispiel an Personen ausgeliehen werden. Zu jedem Zeitpunkt kann ein Asset maximal einer Person zugeordnet werden, wodurch das mehrfache oder gleichzeitige Ausleihen eines Assets verhindert wird. Darüber hin-

aus beschreiben Status-Label den Zustands eines Assets und ob dieses ausgeliehen werden kann. Alle Funktionalitäten können zudem über eine REST-API programmiert genutzt werden. Des Weiteren verfügt *Snipe-IT* über eine Schnittstelle, welche die Integration von LDAP stark vereinfacht.

#### Level 2: Container

Im zweiten Level werden die Container des Software-Systems gezeigt. Hierbei werden Verantwortlichkeiten und die Kommunikation zwischen den Bestandteilen des Software-Systems dargestellt (Abbildung 3.2)(Brown, 2021).

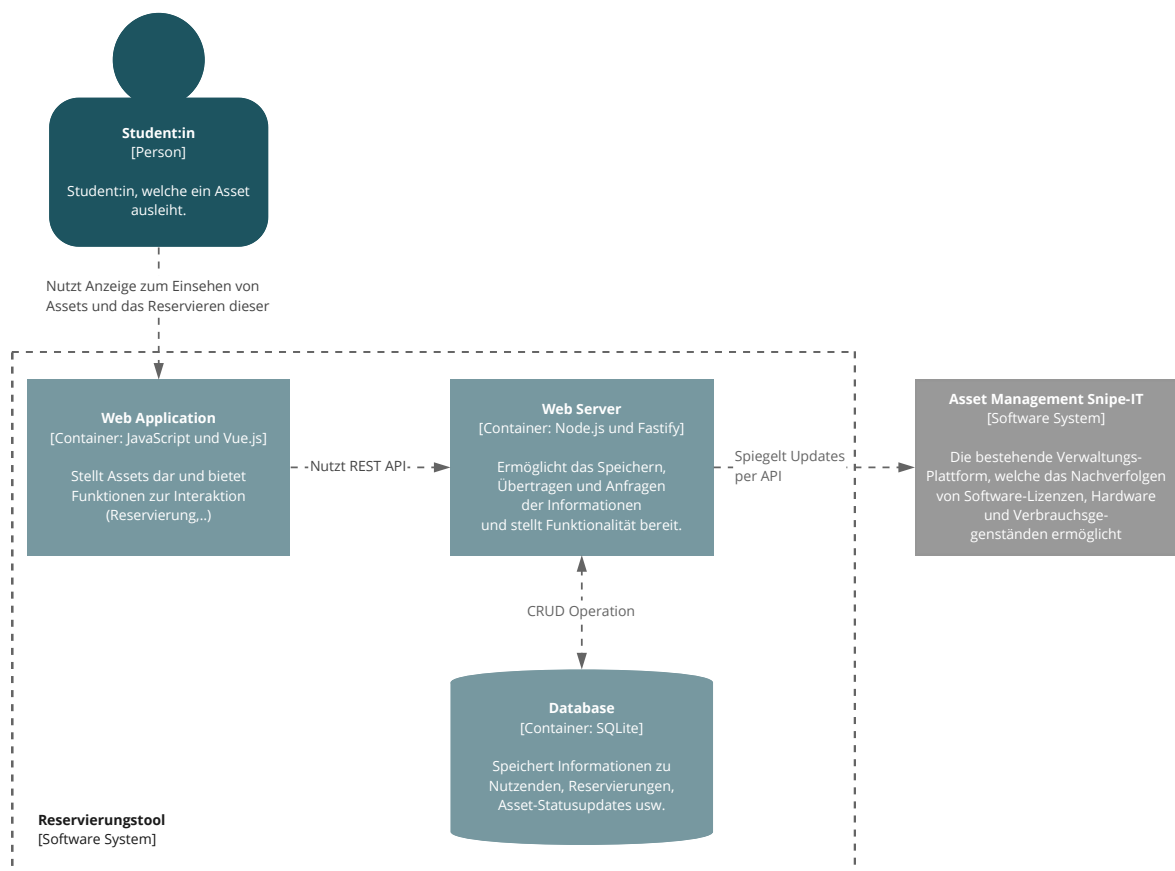


Abbildung 3.2: Architektur Level 2 nach Brown (2013)

Für die Grundlage des Frontends wird Vue.js<sup>1</sup> verwendet. Vue.js ist ein progressiver JavaScript Framework. Durch das Vite<sup>2</sup> Plugin-System kann eine PWA-Funktionalität mithilfe

<sup>1</sup><https://vuejs.org/>

<sup>2</sup><https://vitejs.dev/>

des `vite-plugin-pwa`-Plugins<sup>3</sup> schnell eingebunden werden. Außerdem zeigt Vue.js in den Leistungs-Benchmarks positive Merkmale. Zusätzlich zu den bereits aufgeführten Eigenschaften wird Vue.js aufgrund der begrenzten Implementierungszeit und bestehenden Erfahrung gewählt.

(Krause,  
2022)

Das Reservierungsinterface nutzt die von Snipe-IT bereitgestellten Daten. Die Hauptaufgabe der Schnittstelle ist das Reservieren von Assets in die Zukunft. Da Snipe-IT selbst die zukünftige Reservierung nicht unterstützt, werden diese Reservierungen stattdessen im Reservierungsinterface gespeichert. Die gesamte Kommunikation zwischen Frontend und Snipe-IT wird somit über das Reservierungsinterface stattfinden, um auch alle zukünftigen Reservierungen zu berücksichtigen. Die Datenbank und API von Snipe-IT bilden die Ausgangsposition des Systems. Folglich wird die Verwaltung und Speicherung der vorhandenen Assets über Snipe-IT direkt abgewickelt. Über die bereitgestellte API werden die gespeicherten Daten für alle weiteren Komponenten bereitgestellt.

### Level 3: Components

Das dritte Level stellt die Container aus Level 2 genauer dar, um die elementaren, strukturellen Bestandteile und Wechselwirkungen zwischen diesen aufzuzeigen (Brown, 2021). Im Folgenden werden die Container *Web-Application* und *Web Server* genauer betrachtet (Abbildung 3.3).

Für das Reservierungsinterface wird ein Serverframework und eine Speichermöglichkeit in Form einer Datenbank benötigt. Aufgrund der ausgeprägten Pluginauswahl<sup>4</sup> und breiten Nutzung<sup>5</sup> wird Fastify<sup>6</sup> als Serverframework eingesetzt. Als relationale Datenbank wird die quelloffene Software SQLite<sup>7</sup> eingesetzt, da diese zur Verwendung keine komplexe Einrichtung benötigt. Im Gegensatz zu Datenbanken wie *PostgreSQL* oder *MySQL* benötigt SQLite keine Installation eines Datenbank-Servers. Alle Daten werden in einer alleinstehenden Datei gespeichert (<https://www.sqlite.org/different.html>). Trotz der simplen Einrichtung bietet SQLite ein umfangreiches Sortiment an SQL-Funktionen und unterstützt auch den Umgang mit großen Datensätzen (<https://www.sqlite.org/limits.html>). Um den Zugriff und die Verwaltung der Daten zu vereinfachen, wird zudem die ORM-Bibliothek Prisma<sup>8</sup> verwendet.

v-calender Komponente erläutern

<sup>3</sup><https://github.com/vite-pwa/vite-plugin-pwa>

<sup>4</sup><https://www.fastify.io/ecosystem/>

<sup>5</sup><https://www.fastify.io/organisations/>

<sup>6</sup><https://www.fastify.io/>

<sup>7</sup><https://www.sqlite.org/index.html>

<sup>8</sup><https://www.prisma.de/>

### 3 Konzeption

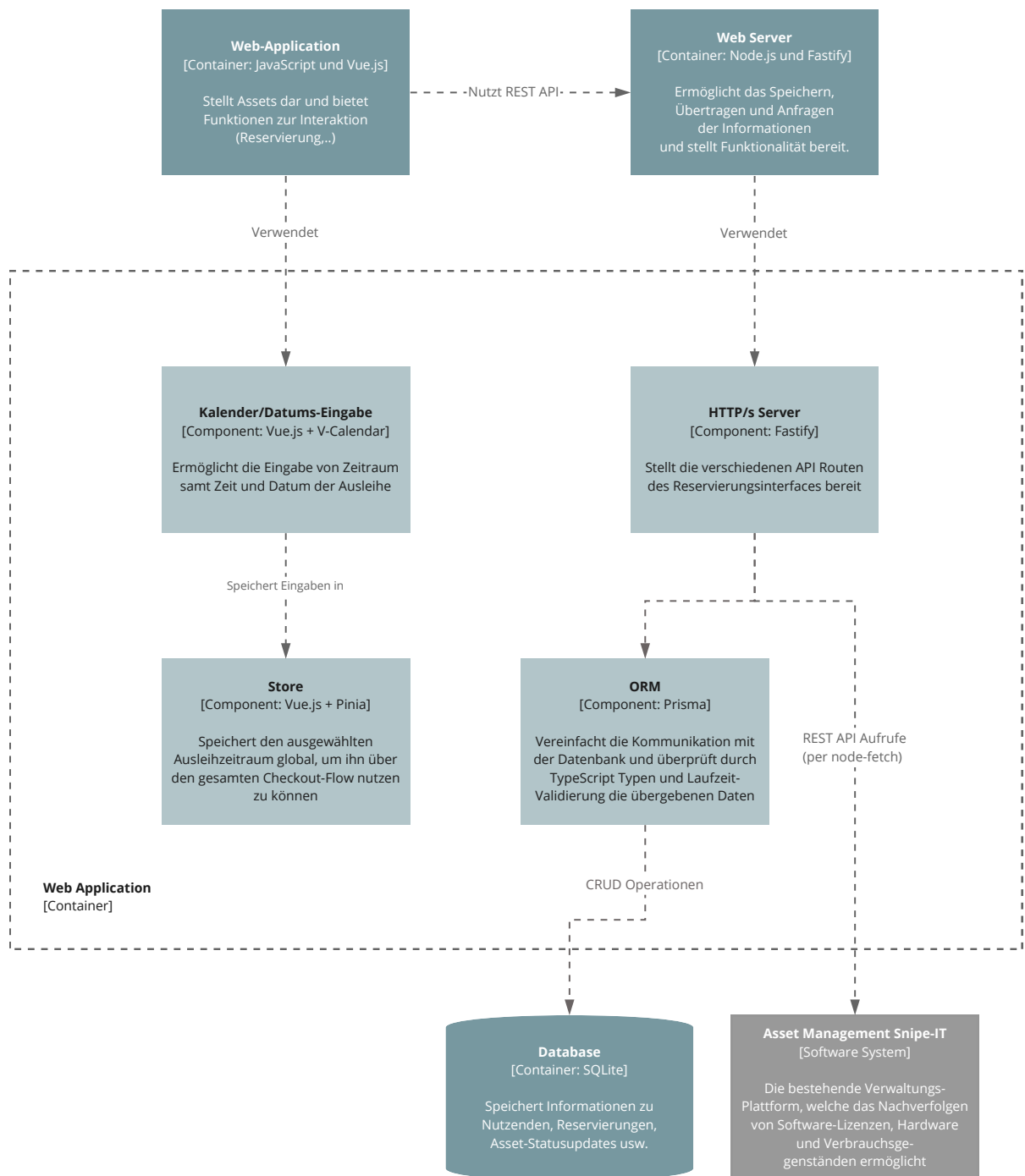


Abbildung 3.3: Architektur Level 3 nach Brown (2013)



# 4

## Interface-Design

Neben den inhaltlichen Funktionalitäten ist im menschenzentrierten Gestaltungsprozess vor allem das Interface-Design und die Gestaltung der Komponenten von zentraler Bedeutung (DIN EN ISO 9421-210:2020-03, 2020) (DIN EN ISO 9241-210, 2011). Bei der Konzeption des Designs sollten die erarbeiteten Funktionalitäten aus Kapitel 3 berücksichtigt werden. Folglich sollten die in Abschnitt 2.2 erarbeiteten Nutzenden die Gestaltungslösungen verstehen und bedienen können.

Die Medieninformatik bietet viele Methoden und Vorgehensweisen zur menschenzentrierten Gestaltung (Herczeg, 2009b). Im Rahmen der Arbeit wurde vor allem Prototyping in verschiedenen Formen eingesetzt (Herczeg, 2009a). Des Weiteren wurden die Zehn Usability Heuristiken nach Nielsen (1994a) bei der Entwicklung stets beachtet, um Kriterien der Gebrauchstauglichkeit zu adressieren. Eine ausführliche Aufzählung der Heuristiken befindet sich in Anhang B. In der Gestaltung wurde nach der »Mobile First«-Strategie gearbeitet (/R40/). Hierbei wird die Oberfläche zunächst für kleine Bildschirmflächen entwickelt, um die Informationsdichte an diesen zu orientieren. Erst beim Erweitern der Fläche werden zusätzliche Informationen entsprechend des verfügbaren Platzes angezeigt (Kim, 2013). Bei der Anordnung der Komponenten wurde sich an den in Unterabschnitt 2.1.1, genannten Anwendungen, wie zum Beispiel Airbnb<sup>1</sup> und Otto<sup>2</sup> orientiert. Des Weiteren wurden Tailwind UI<sup>3</sup> und Headless UI<sup>4</sup> als Inspirationsquelle herangezogen, um die Realisierung des Systems zu erleichtern.

Die erste Iteration des Interface-Designs umfasst Skizzen. Die Skizzen wurden erstellt, um einen groben Überblick der verschiedenen Ansichten zu erhalten und in *Mockup*<sup>5</sup> umge-

---

<sup>1</sup><https://www.airbnb.de/>

<sup>2</sup><https://www.otto.de/>

<sup>3</sup><https://tailwindui.com/components>

<sup>4</sup><https://headlessui.com/>

<sup>5</sup><https://getmockup.app/>

setzt. Während des Designprozesses wurden die Skizzen mit Figma<sup>6</sup> zu einem High-Fidelity-Prototyp weiterentwickelt.

Im Rahmen dieses Kapitels werden die zentralen Ansichten und Design-Entscheidungen erläutert. Eine Übersicht aller Entwürfe befindet sich im .

### 4.1 Iteratives Vorgehen

Für die Tests wurden Usability-Spezifikationen aus Szenarien abgeleitet. Die Szenarien beschreiben Aufgaben, welche als Vorlage in der Evaluation dienen können . Um repetitive Evaluationsergebnisse zu verhindern, wurden die Evaluationsergebnisse kontinuierlich in die weitere Entwicklung eingearbeitet. Tabelle 4.1 stellt die Evaluationsteilnehmenden mit IDs dar, welche in den folgenden Abschnitten als Verweise verwendet werden.

**Tabelle 4.1:** Teilnehmende der Zwischenevaluation

ID	Alter	Rolle
E1	19 - 25 J.	Medieninformatiker:in
E2	19 - 25 J.	Robotiker:in
E3	19 - 25 J.	Medieninformatiker:in, Hilfwissenschaftler:in
E4	19 - 25 J.	Medieninformatiker:in
E5	19 - 25 J.	Medieninformatiker:in, Hilfwissenschaftler:in
E6	25 - 30 J.	Wissenschaftliche:r Mitarbeiter:in

### 4.2 Designsprache

Im Folgenden wird auf die Entwicklung der wichtigsten Komponenten und Ansichten in der Interfacegestaltung eingegangen.

<sup>6</sup><https://www.figma.com/de/>

### 4.2.1 Wortlaut

Ein zentrales Problem des Prototyps war der konsistente Wortlaut (H2, H4, H5). Insbesondere der Begriff *Assets* konnte mehrfach nicht oder schlecht interpretiert werden (E1, E2). Daraufhin wurden Vorschläge geliefert, wie *Material*, *Hardware*, *Systeme*, *Geräte*. Im weiteren Verlauf hat sich *Material* als präferierter Begriff herausgestellt (E3-E5). Des Weiteren hat das *Suchen nach Kriterien* für Missverständnisse in Verbindung mit der ebenfalls existierenden *Suche* ergeben (E1, E2). Daher wurden auch hier verschiedene Begriffe wie *Auswahlhilfe*, *Suchhilfe*, *Kriterien-Suche*, *Kriterien-Hilfe*, *Auswahl nach Kriterien*, *Ausleih-Hilfe* erarbeitet. Da die Funktion im Rahmen dieser Arbeit nicht realisiert wurde, ist die Entscheidung bei diesen Begriffen ausstehend.

Ausblick

### 4.2.2 Schrift

Allgemein wurde die Schriftart nach Kriterien für eine gute Lesbarkeit, wie: Erkennbarkeit, Unterscheidbarkeit und Offenheit betrachtet (leserlich.info, 2022). Außerdem wurde sich für die spätere Realisierung der Typografie direkt an Tailwind UI<sup>7</sup> orientiert und somit Segoe UI gewählt (Abbildung 4.1).

Class	Properties	
<code>text-xs</code>	<code>font-size: 0.75rem; /* 12px */</code> <code>line-height: 1rem; /* 16px */</code>	<code>text-sm</code> The quick brown fox jumps over the lazy dog.
<code>text-sm</code>	<code>font-size: 0.875rem; /* 14px */</code> <code>line-height: 1.25rem; /* 20px */</code>	<code>text-base</code> The quick brown fox jumps over the lazy dog.
<code>text-base</code>	<code>font-size: 1rem; /* 16px */</code> <code>line-height: 1.5rem; /* 24px */</code>	<code>text-lg</code> The quick brown fox jumps over the lazy dog.
<code>text-lg</code>	<code>font-size: 1.125rem; /* 18px */</code> <code>line-height: 1.75rem; /* 28px */</code>	<code>text-xl</code> The quick brown fox jumps over the lazy dog.
<code>text-xl</code>	<code>font-size: 1.25rem; /* 20px */</code> <code>line-height: 1.75rem; /* 28px */</code>	<code>text-2xl</code> The quick brown fox jumps over the lazy dog.

Abbildung 4.1: Typografie von Tailwind UI

<sup>7</sup><https://tailwindui.com/components>

### 4.2.3 Farbschema

Die Farben der Anwendung (Abbildung 4.2) werden nach dem 60-30-10 Prinzip genutzt (Gordon, 2021). Hierbei stellt grau die Hauptfarbe dar, weiß die Sekundärfarbe und orange die Akzentfarbe. Da die Anwendung für den Gebrauch am IMIS entwickelt wird und die Farben des Instituts sich auf Orange und das Universitäts-Blau beschränken, wurde sich nach einigen Vergleichen für das Orange entschieden (E4,E5). Der dunkle Ton stellt die Textfarbe der Anwendung dar.



**Abbildung 4.2:** Farbsystem des Interface-Designs

### 4.2.4 Dashboard

Positiv wurde stets angemerkt, dass die Anwendung im allgemeinen verständlich und nicht überfordernd wirkt (H8) (E1-6). Aufgrund des zu Beginn geringen Inhalts auf der Startseite ist eine ansprechende und fokussierte Anordnungen der Komponenten wichtig (H8). Abbildung 4.3 zeigt die Entwicklung des Dashboards. Ein wichtiger Punkt bildet die im linken Bild fehlende Orientierung (E1-4). Außerdem wirkte die Ansicht durch die viele einzelne Elemente schnell überladen (H8) (E1-2). Als Konsequenz wurde sich für eine Tab-Leiste entschieden. Folglich ist die Seite auch bei mehr als drei Reservierungen übersichtlich (E4). Sobald ein Asset ausgeliehen wurde, sollten die entsprechenden Informationen (Status, Zeitraum, Ort) übersichtlich und schnell eingesehen werden können (H1) (F-A-1). Dies ermöglicht die Kartenansicht, wobei eine Möglichkeit zum Löschen und das Bearbeiten des Zeitraums erwünscht war und in der Realisierung integriert wurde (E6).

Die Verwaltungsansicht, in welcher die Möglichkeit zum Aktualisieren des Status besteht, entspricht zum Großteil dem Dashboard (H6) (F-V-1, F-V-2).

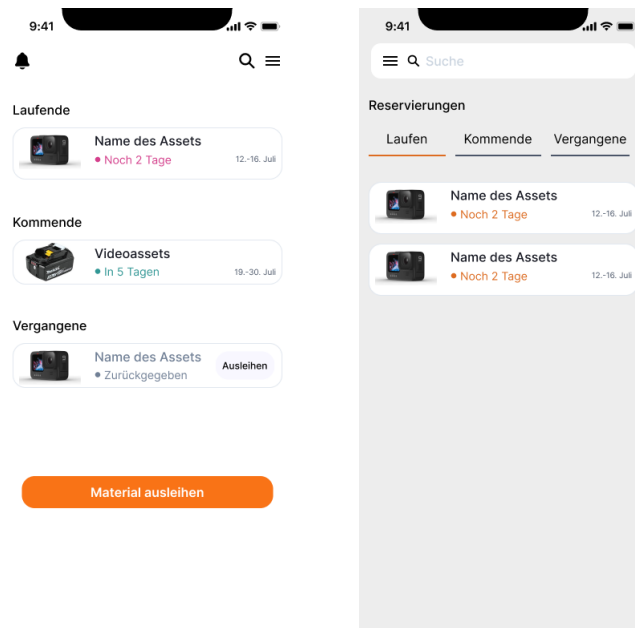
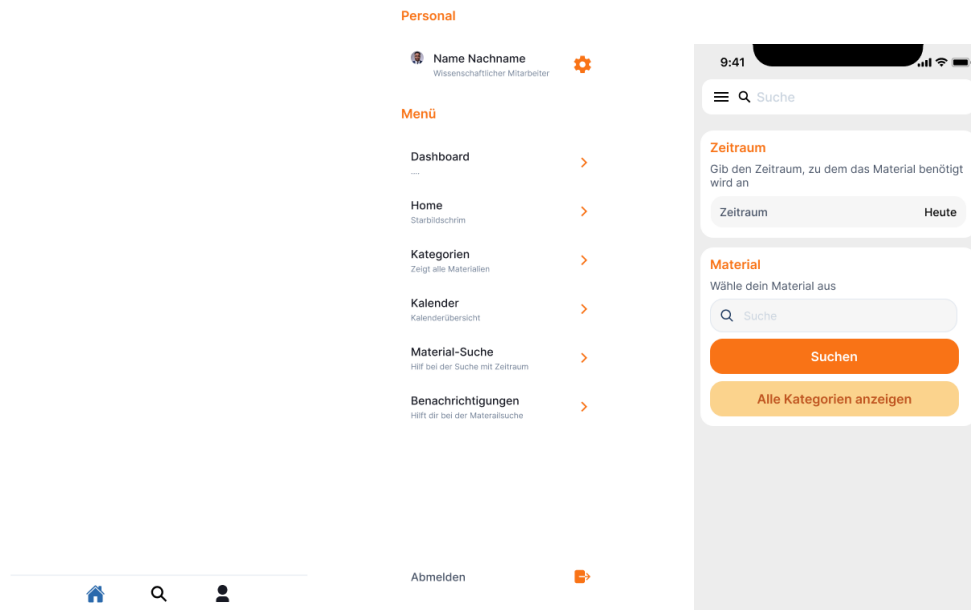


Abbildung 4.3: Dashboard in der mobilen Ansicht

#### 4.2.5 Navigation und Suchleiste

In der Interface-Gestaltung gibt es verschiedene Möglichkeiten, die Navigation bereitzustellen. Für den Mobilien Kontext wurde eine Navigationsleiste am unteren Bildschirmrand und ein Burgermenü in Erwägung gezogen. Für die Desktop-Navigation wurde zwischen einer festen Navigationsleiste links und einer Leiste oben abgewägt (Abbildung 4.4). Am Ende wurde sich für eine Navigationsleiste auf der linken Seite entscheiden. Orientiert wurde sich hierbei an bekannten Anwendungen mit ähnlichen Funktionalitäten und dem Material Design (Google, 2022). Die Umgangsweise mit diesen Gestaltungslösungen ist bereits bekannt und somit übertragbar (H4, H6). Daran anknüpfend wurde sich für eine integrierte Suchleiste entschieden, sodass Nutzende jederzeit die Möglichkeit haben, nach Assets zu suchen (Google, 2022) (E4).



**Abbildung 4.4:** Navigationsmöglichkeiten und Suchleiste

### 4.2.6 Kategorien und Suche

Die Übersicht der Kategorien wurde ebenfalls an bekannte Anwendungen mit ähnlichen Funktionen angelehnt (H4, H8). Dies führt unter anderem dazu, dass Fehler besser vermieden werden können (H9).

Die Suche wurde der Anwendung Airbnb<sup>8</sup> nachempfunden, da viele Nutzende den Reservierungsvorgang mit dieser Anwendung assoziieren (H2, H4). Die Suche sollte durch das direkte Einstellen eines Ausleihzeitraums Fehler vorbeugen und die Verfügbarkeit anzeigen (H5) (F-VA-6). Abbildung 4.5 zeigt die erste Version des High-Fidelity-Prototypen, bei welcher ein Suchbutton fehlte. Zudem wurden die Vorschläge der angezeigten Kategorien weniger genutzt und direkt auf die Suche oder *alle Kategorien Anzeigen* geklickt. Daher wurden diese Elemente entsprechend angepasst (E1, E2, E4).

Für die Unteransicht wurde sich für eine Kartenansicht entschieden (Abbildung 4.6). Diese ergab sich aus dem bereits bekannten Vorgängen und Interfaces von Online-Einkäufen. Nutzenden wird mithilfe eines Bildes direkt gezeigt, um was für ein Asset es sich handelt. Außerdem wird auch hier die Assetverfügbarkeit mit eingebunden (H4, H8) (F-VA-3).

<sup>8</sup><https://www.airbnb.de/>

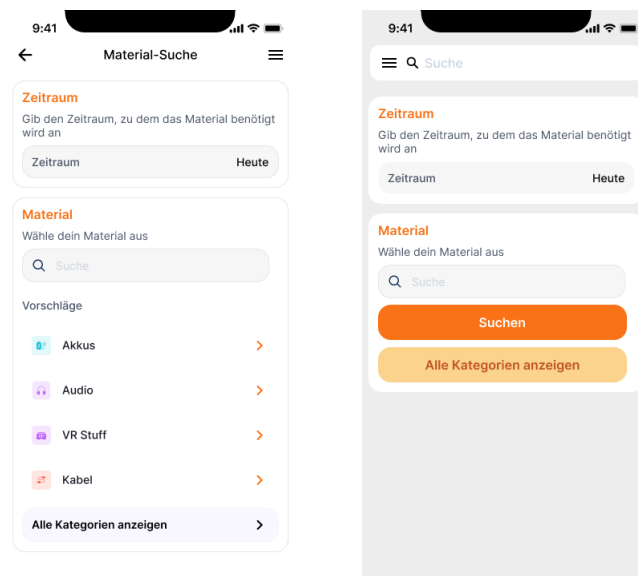


Abbildung 4.5: Materialsuche

### 4.2.7 Kalender

Die Kalenderkomponente wurde lediglich als Skizze veranschaulicht (Abbildung 4.7). Im High-Fidelity-Prototyp wurde sich bereits an der Komponente *V-Calendar*<sup>9</sup> orientiert, welche die wichtigsten Funktionen bereits implementiert und so die Realisierung erleichtert.

Für den Kalender unter anderem die Möglichkeit zur Ausblendung von Wochenendtagen wichtig, da zu diesen Zeitpunkten keine Ausleihe möglich ist (H1) (F-VA-3). Außerdem sollten vergangene Tage nicht auswählbar sein. Weitere Punkte wurden auch hier wiederholt von Airbnb beachtet (H4, H7, H8) (E2-4).

Für Verleihende ist eine generelle Übersicht über einzelne Tage wichtig, um Abholungen und Rückgaben von Assets besser planen zu können (F-V-4).

<sup>9</sup><https://vcalendar.io/layouts.html>

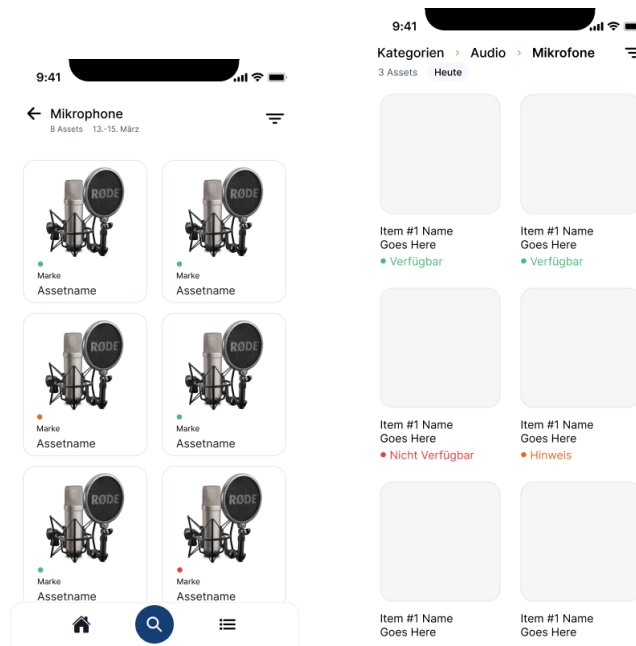


Abbildung 4.6: Materialansicht (Kacheln)

### 4.2.8 Asset-Detailansicht und Reservierungen

Die Ansicht der einzelnen Assets sollte insbesondere die Assetverfügbarkeit, Funktionalitäten des Assets und die zuständige Person enthalten (F-VA-3, F-VA-4, F-VA-8). Abbildung 4.8 zeigt die genannten Informationen. Es ergab sich, dass eine Assetbeschreibung des Artikels zunächst nicht von Bedeutung sei (E6). Den Ausleihzeitraum separat zur Abholung und Rückgabe anzeigen zu lassen, führte im Verlauf häufig zu Verwirrung. Daher wurden die zeitlichen Daten auf die Abholung und Rückgabe beschränkt (H5) (E5, E6).





Abbildung 4.7: Kalenderkomponente

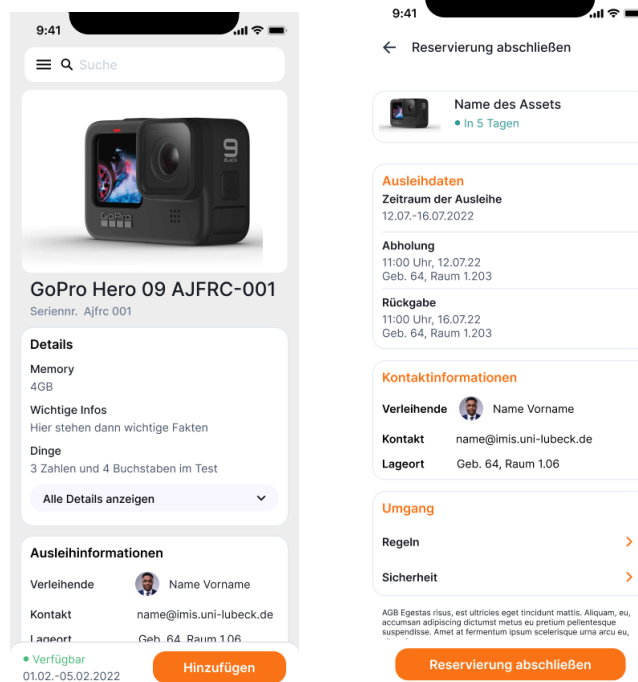


Abbildung 4.8: Reservierung und Check-out

# 5

## Implementierung

Das folgende Kapitel beschreibt die Implementierung des Reservierungsinterfaces sowie des Frontends. Zunächst wird die Implementierung des Reservierungsinterfaces und die damit einhergehenden technischen Aspekte beschrieben. Dabei wird Aufschluss über die Struktur gegeben und die Kernfunktionalitäten sowie Herausforderungen in der Realisierung näher erläutert. Daraufhin wird die Umsetzung des Frontends erläutert. Abschließend wird auf die Inbetriebnahme des Systems eingegangen.

### 5.1 Implementierung des Reservierungsinterfaces

Dieser Abschnitt erläutert den technischen Aufbau des Reservierungsinterfaces und geht auf relevante Aspekte in der Realisierung der Kernfunktionalitäten (Unterabschnitt 2.6.4) ein. Des Weiteren werden unerwartete Herausforderungen thematisiert, welche im Rahmen der Arbeit nicht bewältigt werden konnten.

#### 5.1.1 Aufbau des Reservierungsinterface

Das Reservierungsinterface teilt sich in drei wesentliche Bestandteile: der *Fastify-HTTP-Server*, die *SQLite Datenbank* und das *ORM Prisma* (vgl. Abbildung 3.3). Diese Komponenten spiegeln sich auch in der Verzeichnisstruktur aus Abbildung 5.1 wider.

Der HTTP-Server findet sich in der *server.ts* wieder und stellt dort die API des Reservierungsinterfaces bereit. Die entwickelte API lässt sich in drei Bereiche teilen: *Assets*, *Kategorien* und *Reservierungen* (Tabelle 5.1). Für die drei Inhaltstypen werden die Routen aus Tabelle 5.1 bereitgestellt, welche die von Fielding und Reschke (2014) beschriebene Semantik für HTTP-Methoden beachten. Folglich werden bei Verwendung der *GET*-Methode

```

prisma
├── migrations
│   ├── 20220904101851_init
│   │   └── migration.sql
│   ├── 20220913131925_add_returned_and_received
│   │   └── migration.sql
│   └── migration_lock.toml
├── dev.db
└── schema.prisma

```

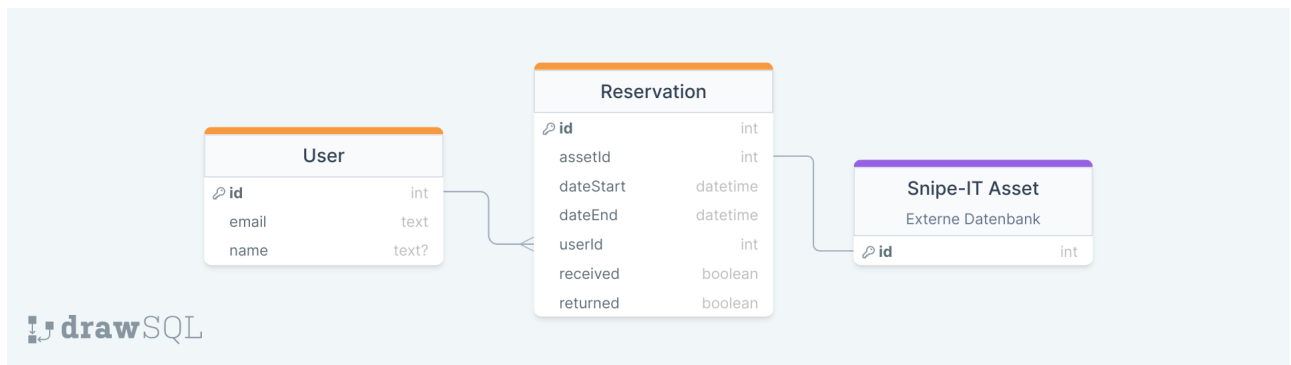
**Abbildung 5.1:** Verzeichnisstruktur des Reservierungsinterfaces

ausschließlich Daten zurückerhalten. Hingegen muss bei einer Anfrage mit zu übermittelnden Daten die *POST*-Methode verwendet werden, um einen neuen Eintrag im System zu erschaffen. Beispielsweise wird eine *GET*-Anfrage an */assets/:id* abgeschickt, um die Informationen eines Assets zu erhalten. Um den Status in Snipe-IT auf *herausgegeben* zu aktualisieren, wird eine *POST*-Anfrage an */reservation/receive* gesendet, sobald Verleihende eine abgeholte Reservierung bestätigen.

**Tabelle 5.1:** API des Reservierungsinterfaces

Methode	Route	Funktion
GET	<i>/assets</i>	Erhalte alle Assets
GET	<i>/assets/:id</i>	Erhalte ein Asset mit der entsprechende ID
GET	<i>/categories</i>	Erhalte alle Kategorien
GET	<i>/reservation</i>	Erhalte Reservierungen
POST	<i>/reservation</i>	Erstellen Reservierung
POST	<i>/reservation/receive</i>	Erstellen Reservierung
POST	<i>/reservation/return</i>	Erstellen Reservierung
POST	<i>/reservation/id</i>	Erstellen Erstellen Reservierung
DELETE	<i>/reservation/delete</i>	Löschen Reservierungen
PATCH	<i>/reservation/patch</i>	Verändern Reservierung

Die mit SQLite bereitgestellte Datenbank speichert die Reservierungen sowie die Profile. Hierfür wurde das in Abbildung 5.2 dargestellte Datenbankschema erarbeitet. Zur Umsetzung dieses Schemas wurde das ORM Prisma genutzt, welches drei zentrale Funktionen bietet.



**Abbildung 5.2:** Datenstruktur der Reservierungen in Verbindung mit Nutzenden

1. Beschreibung des Schemas durch eigene Domain Specific Language (DSL) (Abbildung 5.3)
2. Automatisierte Migration bei Schemaänderung
3. JavaScript-Client mit generierten TypeScript-Typen (Abbildung 5.4)

### 5.1.2 Implementierung der Kernfunktionalität

Dieser Abschnitt präsentiert die Implementierung der Kernfunktionalität des Reservierungsinterfaces, welche aus den Anforderungen bestimmt wurde (Abschnitt 2.6). Bei der Funktionalität handelt es sich um das Reservieren in die Zukunft, sowie das Speichern dieser Vorgänge und die damit einhergehende Bestätigung für die Aktualisierung in Snipe-IT.

Sobald ein Asset über die Web-App ausgeliehen wird, werden die relevanten Daten mit einer *POST*-Anfrage an die */reservation*-Route des Reservierungsinterfaces gesendet. Diese relevanten Daten (Assetname, Datum, Uhrzeit, Ort, etc.) werden im Reservierungsinterfaces gespeichert. Sobald Verleihende die Ausleihe bestätigen, wird der Ausleihstatus mit einer *POST*-Anfrage an */reservation/receive* intern und in Snipe-IT von *ausleihbar* zu *herausgegeben* aktualisiert. Zur Kommunikation zwischen dem Reservierungsinterface und Snipe-IT wird das Paket *node-fetch* genutzt, welches die Fetch-API des Browsers in *Node.js* bereitstellt. Für das Aktualisieren des Status eines Assets mit der *assetId* wird beispielsweise eine *POST*-Anfrage an `<SnipeIt_URL>/api/v1/hardware/<assetId>/checkout` versendet, welche die ID des *herausgegeben*-Status und die ID des ausleihenden Nutzenden beinhaltet (Abbildung 5.5). Analog geschieht dies für das Zurückgeben eines Assets mit einer *POST*-Anfrage an */reservation/return*.

```
schema.sql

1  generator client {
2    provider = "prisma-client-js"
3  }
4
5  datasource db {
6    provider = "sqlite"
7    url      = env("DATABASE_URL")
8  }
9
10 model User {
11   id      Int      @id @default(autoincrement())
12   email   String   @unique
13   name    String?
14   reservations Reservation[]
15 }
16
17 model Reservation {
18   id      Int      @id @default(autoincrement())
19   assetId Int
20   dateStart DateTime
21   dateEnd  DateTime
22   user     User    @relation(fields: [userId], references: [id])
23   userId   Int
24   received Boolean @default(false)
25   returned Boolean @default(false)
26 }
```

**Abbildung 5.3:** Quellcode: schema.prisma

Um auf die Assets zugreifen zu können, wurde mit der Snipe-IT JSON REST API gearbeitet. Die Snipe-IT API umfasst viele Routen<sup>1</sup> für die Abfrage und Manipulation der internen Datenbank, wovon */hardware*, */categories* und */statuslabels* relevant für die Umsetzung dieser Arbeit waren. Abbildung 5.6 zeigt den Aufbau und die Beziehungen der genutzten Snipe-IT API Ressourcen, wobei die Ressource *hardware* den Assets in der Snipe-IT Datenbank entspricht.

Um mit der API arbeiten zu können, muss ein *persönliches Zugriffstoken*<sup>2</sup> für die Nutzung generiert werden. Das persönliche Zugriffstoken muss mit jeder Anfrage an die API, in Form eines Bearer <Token> HTTP-Headers, mitgeschickt werden. Da persönliche Zugriffstoken verwendet werden, spiegeln die Berechtigungen des API-Tokens die Berechtigungen des Nutzenden wider.

<sup>1</sup><https://snipe-it.readme.io/reference/api-overview>

<sup>2</sup><https://snipe-it.readme.io/reference/generating-API-tokens>

```

migration.sql

1 PRAGMA foreign_keys=OFF;
2 CREATE TABLE "new_Reservation" (
3     "id" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
4     "assetId" INTEGER NOT NULL,
5     "dateStart" DATETIME NOT NULL,
6     "dateEnd" DATETIME NOT NULL,
7     "userId" INTEGER NOT NULL,
8     "received" BOOLEAN NOT NULL DEFAULT false,
9     "returned" BOOLEAN NOT NULL DEFAULT false,
10    CONSTRAINT "Reservation_userId_fkey" FOREIGN KEY ("userId")
    REFERENCES "User" ("id") ON DELETE RESTRICT ON UPDATE CASCADE
11 );
12 INSERT INTO "new_Reservation" ("assetId", "dateEnd", "dateStart",
    "id", "userId") SELECT "assetId", "dateEnd", "dateStart", "id",
    "userId" FROM "Reservation";
13 DROP TABLE "Reservation";
14 ALTER TABLE "new_Reservation" RENAME TO "Reservation";
15 PRAGMA foreign_key_check;
16 PRAGMA foreign_keys=ON;

```

Abbildung 5.4: Quellcode: migration.sql

### 5.1.3 Herausforderungen in der Einbindung von LDAP

Snipe-IT bietet bereits eine integrierte Lösung zur Synchronisation und Login mit LDAP<sup>3</sup>. Da die *persönlichen Zugriffstoken* jedoch die einzige Authentifizierungsmöglichkeit der API sind und lediglich manuell im Dashboard generiert werden können, kann das LDAP-System der Universität zu Lübeck nicht ohne Umstände eingebunden und entsprechend genutzt werden.

Ein möglicher Lösungsansatz wäre das programmatische Nutzen eines Browsers aufseiten des Reservierungsinterfaces, welcher die Angaben von Nutzenden in der Web-App verwendet, um mit diesen den LDAP-Login in Snipe-IT durchzuführen. Sollte der Snipe-IT Login erfolgreich sein, könnten Nutzende für den Rest der Sitzung als authentifiziert gelten. Im Reservierungsinterface selbst würde ein speziell für das Reservierungsinterface generierte Token für die Kommunikation mit Snipe-IT genutzt werden.

<sup>3</sup><https://snipe-it.readme.io/docs/ldap-sync-login>

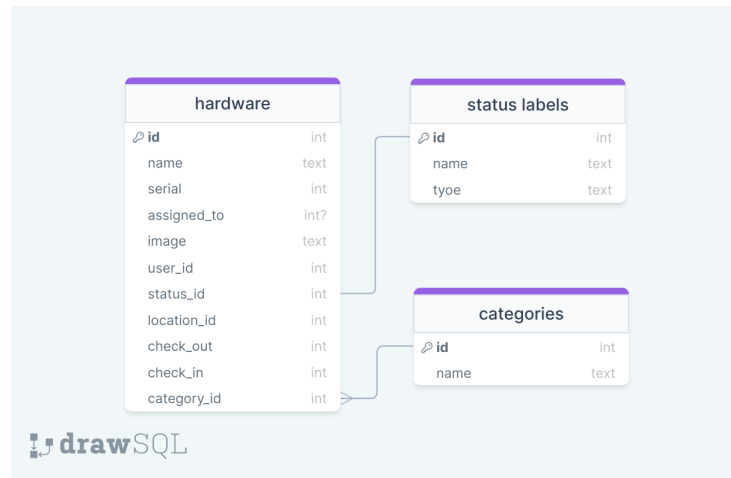


Abbildung 5.5: UML Snipe-IT API

## 5.2 Implementierung des Frontends

Das kommende Kapitel beschreibt die Client-seitige Realisierung der Arbeit. Zunächst wird der Aufbau betrachtet, daraufhin wird auf das Extrahieren der Unterkategorien eingegangen (Abbildung 5.7).

Für den Aufbau des Projektes wurde aus den in Kapitel 3 festgestellten Anforderungen *vue.js* verwendet. Bei der Implementierung wurde sich an den best practices der *Vue.js*-Dokumentation orientiert. Für sich wiederholende Elemente wurden eigene Views erstellt. Dadurch ergibt sich eine hierarchisch geschachtelte Client-Anwendung der Vue-Komponenten. Abbildung 5.7 stellt die Komponenten-Struktur vereinfacht dar. Um konkretere Vorschläge in der Entwicklungsumgebung zu ermöglichen und vorzeitige Fehler zu minimieren, wurde ergänzt zu *JavaScript TypeScript* verwendet.

(Vue.js,  
2021a)

### 5.2.1 Aufbau der Routen und Komponentenstruktur

Für die Realisierung der Routen wurde der File-Based-Routing Ansatz verwendet, welcher die Routen aufgrund der Verzeichnisstruktur generiert. Hierfür wurde das Vite-Plugin *vite-plugin-pages*<sup>4</sup> genutzt. Vue-Komponenten, welche sich im */pages*-Verzeichnis des Projekts befinden, werden in eine, dem Dateinamen gleichende, Route umgewandelt. Ordner können hierbei verwendet werden, um Unterroueten zu erstellen, während die Verwendung von eckigen Klammern zur Kennzeichnung von Parametern genutzt wird. Beispielsweise wird

<sup>4</sup><https://github.com/hannoeru/vite-plugin-pages>

Der  
vorherige  
Statz=DOOF



```

server.ts

1  const asset = await fetch(
2    process.env.SNIPEIT_URL +
3    '/api/v1/hardware/' +
4    reservation.assetId +
5    '/checkout',
6    {
7      ...options,
8      method: 'post',
9      body: JSON.stringify({
10        checkout_to_type: 'user',
11        status_id: 4,
12        assigned_user: 14,
13      }),
14    }
15 )

```

Abbildung 5.6: Quellcode: server.ts

eine *[id].vue*-Datei, welche sich in dem Ordner *categories* befindet, in die */categories/<id>*-Route umgewandelt, welche die angegebene ID in der Komponente als Parameter nutzen kann (Abbildung 5.7).

```

prisma
├─ migrations
│  └─ 20220904101851_init
│     └─ migration.sql
│  └─ 20220913131925_add_returned_and_received
│     └─ migration.sql
├─ migration_lock.toml
├─ dev.db
└─ schema.prisma

```

Abbildung 5.7: Verzeichnisstruktur des Clients

### 5.2.2 Extrahieren von Unterkategorien

Die im Rahmen dieser Arbeit verwendete Beispieldatenbank beinhaltete eine Vielzahl an Kategorien und Unterkategorien, welche nach folgendem Format benannt wurden: »Kategorie - Unterkategorie«. Das verwendete Format ergibt sich aus der fehlenden Funktionalität Snipe-Its, einem Asset mehrere Kategorien zuweisen zu können. Da das Anzeigen der Kategorien in diesem Format unübersichtlich ist, sollten zunächst nur die Oberkategorien und anschließend die Unterkategorien angezeigt werden. Abbildung 5.8 stellt eine vereinfachte Form des verwendeten Quellcodes dar.



```
server.ts

1  const asset = await fetch(
2    process.env.SNIPEIT_URL +
3    '/api/v1/hardware/' +
4    reservation.assetId +
5    '/checkout',
6    {
7      ...options,
8      method: 'post',
9      body: JSON.stringify({
10        checkout_to_type: 'user',
11        status_id: 4,
12        assigned_user: 14,
13      }),
14    }
15  )
```

**Abbildung 5.8:** Quellcode: server.ts

Zur Extrahierung werden die erhaltenen Kategorien, anhand eines Trennsymbols (»-«), in Ober- und Unterkategoriepaare getrennt. Anschließend werden alle Paare mit der gleichen Oberkategorie zu einem Eintrag zusammengefasst, welcher alle zugehörigen Unterkategorien beinhaltet. Hierbei wird für jede Unterkategorie die hinterlegte Kategorie-ID gespeichert. Somit ergibt sich eine Baumstruktur, welche in der Anwendung schrittweise durchlaufen werden kann.

## 5.3 Nutzung des Systems

Der folgende Abschnitt führt die nötigen Schritte auf, um das System in Betrieb nehmen zu können. Zuerst wird die Installation erklärt, gefolgt von der Konfiguration und Ausführung des Systems.

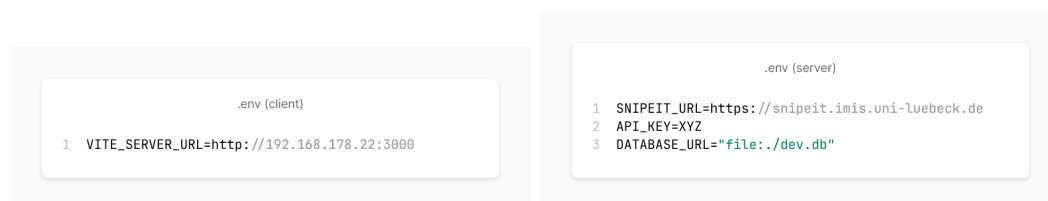
### 5.3.1 Installation

Für die Nutzung des Systems wird eine Installation von *Node.js* vorausgesetzt. Zunächst müssen alle eingebundenen Pakete des *npm* Paketverzeichnis mit `npm install` installiert werden. Dies muss für Server (`\server`) und Client (`\client`) jeweils im entsprechenden Verzeichnis ausgeführt werden. Zusätzlich sollte die Datenbank im Server-Verzeichnis mithilfe des Befehls `npx prisma install` erstellt und vorbereitet werden. Abschließend sollte `npx prisma generate` ausgeführt werden, um die benötigten Dateien des Prisma-Clients zu generieren.

Befehl  
raussuchen

### 5.3.2 Konfiguration

Zur Festlegung von umgebungsabhängigen Variablen werden für Server und Client je eine *.env*-Datei verwendet. Diese müssen vor der Inbetriebnahme angelegt werden und sind nach dem Schema in BILD aufgebaut. Der Server benötigt die URL der zuverwendenden Snipe-IT Instanz, den API-Key zur Authentifizierung (Unterabschnitt 5.3.3) und den Pfad der SQLite Datenbankdatei. Aufseiten des Clients wird lediglich die URL des Reservierungsinterfaces benötigt.



**Abbildung 5.9:** Inhalte der *.env* Datei im Frontend (l) und Backend (r)

### 5.3.3 Einrichtung des Snipe-IT API-Zugangs

Um die gewünschte Datenbank(en) aus Snipe-IT nutzen zu können, wird ein Account in Snipe-IT benötigt, welcher von Administrierenden angelegt werden muss. Daraufhin muss über die Profileinstellung *Manage API Keys* ein API-Key generiert werden. Der generierte API kann nun in der Konfiguration genutzt werden. Außerdem sollte beachtet werden, dass der API Key nur einmalig nach der Generierung einsehbar ist.

### 5.3.4 Ausführung des Reservierungsinterfaces

ausformulieren • Ausführung per `npm run serve`

### 5.3.5 Ausführung der Web-App

Das Ausführen und "bauen" der Web-App geschieht über den Befehl `npm run build-only`. Nachdem der Prozess erfolgreich abgeschlossen ist, sollte ein neues *dist*-Verzeichnis generiert worden sein. Das *dist*-Verzeichnis kann zum statischen Hosting genutzt werden.

## 5.4 Fazit der Implementierung

Die in der Implementierung realisierten Funktionalitäten umfassen das *muss* der in der Formalisierte Anforderungen erarbeiteten Funktionen.

quasi alle Unterüberschriften in einem Satz zusammenfassen, sodass jemand, der das Kapitel nicht gelesen hat, grob ne Idee davon hat, was hier abging

# 6

## Dialogbeispiele

### Bilder überarbeiten und Anordnung

Der folgende Abschnitt präsentiert das in der vorliegenden Arbeit realisierte System anhand eines beispielhaften Nutzungsszenarios. Das Szenario startet mit Mila, einer Erstsemester Studentin im Studiengang Medieninformatik. Mila belegt im ersten Semester das Modul *Einführung in die Medieninformatik (EMI)*. Das Einführungsmodul umfasst eine Gruppenarbeit, in der eine Idee zum Thema *VR/AR* entwickelt werden soll. Am Ende des Semesters soll das Projekt bei den *Media Moments* in der *EMI Award App*<sup>1</sup> ausgestellt werden. Milas Projektgruppe hat sich entschieden eine AR-Anwendung für Erze und Metalle zu gestalten, bei der die Erze und Metalle mithilfe einer App eingescannt werden können und ihre entsprechenden Eigenschaften angezeigt werden. Um das Projekt in der *EMI Award App* präsentieren zu können, will die Gruppe ein Werbevideo für die App aufnehmen.

In einem Videoworkshop von Georg Fink, fiktiver WiMi am IMIS, erfahren Mila und ihre Gruppenmitglieder:innen, dass über die Ausleih-App *Snipe-IT Companion* unter anderem Videoequipment an Studierende verliehen wird.

Mila ruft die Ausleih-App unter der URL <https://snipe-it-companion.de/> auf und meldet sich mit ihren Daten an. Daraufhin wird sie zu einem bisher leeren *Dashboard* weitergeleitet und aufgefordert nach benötigtem Material zu suchen. Da Mila sich nicht sicher ist, welche Kamera sie benötigt, schaut sich Mila unter dem Menü, in den Kategorien um und findet schnell die Kategorie »Kameras«. Nachdem sie auf die Seite der Unterkategorien weitergeleitet wurde, entscheidet sich Mila dafür, eine *GoPro* auszuleihen, weil sie so auch Erze und Metalle unter Wasser aufnehmen können. Dazu klickt Mila auf »Hinzufügen«.

---

<sup>1</sup>RAIMUNDS BA HEHE

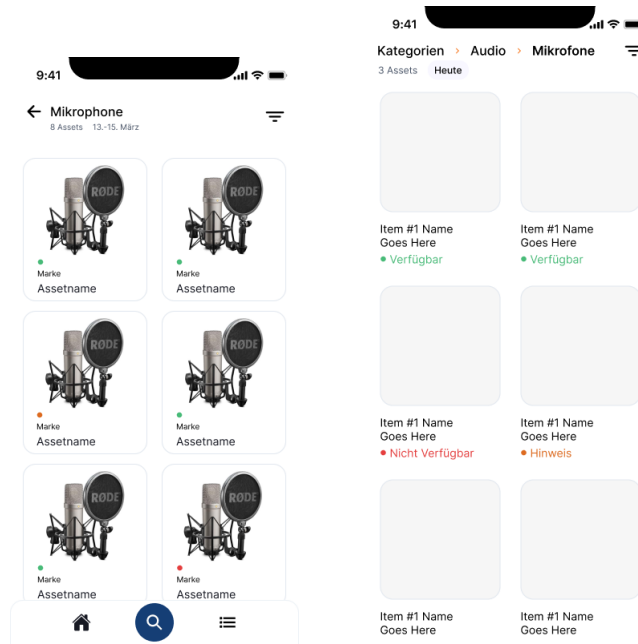
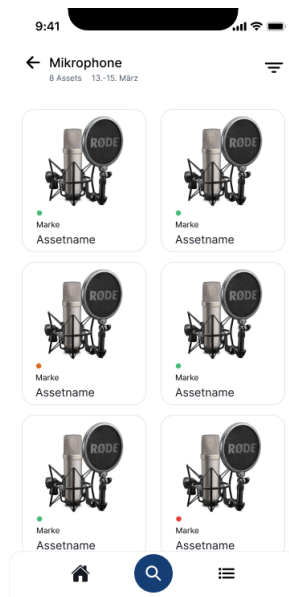


Abbildung 6.1: Kategorien (l), Assetdetails (r)

Nun wird sie dazu aufgefordert, einen Ausleihzeitraum anzugeben. Milas Gruppe hat entschieden, dass sie von kommendem Montag bis Mittwoch das Material benötigten. Da Mila von 8:00 Uhr bis 10:00 Uhr eine Vorlesung hat, gibt sie 10:30 Uhr als Abholzeit und Rückgabezeit an. Um die Reservierung abschließen zu können, klickt Mila auf »Reservieren«. Mila überprüft in der Zusammenfassung, ob ihre Angaben stimmen und stellt fest, dass die Rückgabezeit am Mittwoch doch nicht mehr in ihren Kalender passt. Daher ändert sie die Uhrzeit auf 12:30 Uhr und bestätigt abschließend ihre Reservierung.

Da sich die Gruppe nicht sicher ist, welches Mikrofon für die Aufnahme eines Voiceover sinnvoll ist, suchen sie zunächst in der Ausleih-App nach Mikrofon und sehen, dass Georg Fink für diese zuständig ist. Daraufhin verfasst Mila eine E-Mail an Herrn Fink, in der sie um eine Mikrofon-Empfehlung für Voiceovers bittet. Nachdem dieser mit zwei Vorschlägen geantwortet hat, sucht Mila nach jenen, unter der Angabe ihres Ausleihzeitraums. Direkt stellt Mila fest, dass nur eines der Mikrofone in dem Ausleihzeitraum ausleihbar ist und reserviert dieses.

Georg ist als fiktiver WiMi am IMIS für zwei Labore zuständig, in denen Material ausgeliehen werden kann. Zum Feierabend überprüft er sein Verwaltungsdashboard auf dem Desktop in der Ausleih-App. Er sieht, dass eine neue Reservierung für Montag um 10:30 Uhr von Mila eingegangen ist.



**Abbildung 6.2:** Kategorien (l), Assetdetails (r)

Da sich Mila am Sonntag nicht sicher ist, wo die Materialien abgeholt werden sollen, schaut sie erneut in der Ausleih-App nach und findet den Abholort auf der Dashboardansicht.

Nach der Vorlesung am Montag macht sich Mila auf den Weg in das Gebäude 64 zum Abholort: Techniklabor. Georg wartet dort bereits auf Mila und erklärt ihr, was sie bei der Nutzung der *GoPro* beachten sollte. Nachdem Mila gegangen ist, trägt Georg auf seinem Smartphone ein, dass die Materialien abgeholt wurden. Sobald Georg die Abholung bestätigt hat, wird der Status in der internen Datenbank von Snipe-IT geändert.

Laura Eggers ist ebenfalls fiktive WiMi am IMIS und möchte für eine Studie ein Mikrofon ausleihen. Da sie Zugriff auf Snipe-IT hat, schaut sie zunächst dort nach dem gewünschten Material und stellt fest, dass dieses *herausgegeben* ist. Daraufhin ruft sie die URL <https://snipe-it-companion.de/> auf und reserviert das Mikrofon im gewünschten Zeitraum. Am Abholtag entnimmt sie das Asset und bestätigt die Abholung auf ihrem Smartphone.

Milas Gruppe stellt am Dienstag fest, dass ihnen der Ausleihzeitraum nicht ausreicht und möchte diesen daher um einen Tag verlängern. Dafür öffnet sie die Ausleih-App und sieht auf dem Dashboard unter dem Tab *Laufende* ihre Reservierungen. Daraufhin ändert sie die Daten der beiden Materialien auf Donnerstag um 9:00 Uhr. Am Donnerstag um 9:00 Uhr wartet Mila bereits auf Georg, welcher die Materialien entgegennimmt und in seinem Büro die Rückgabe bestätigt.

Hier wird der Use-Case noch nicht so klar

Noch kurz erwähnen das Georg die Verlängerung auch in seinem Dashboard sehen kann

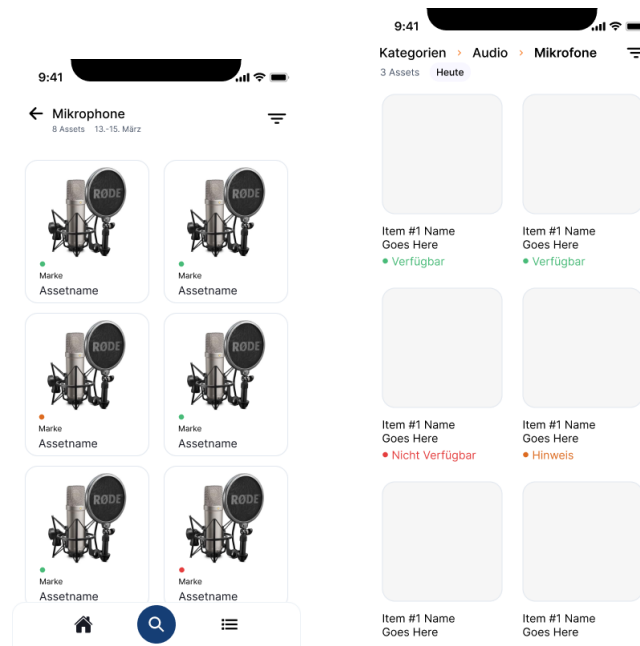


Abbildung 6.3: Kategorien (l), Assetdetails (r)

Im vierten Semester möchte Mila das Mikrofon für das Modul *Interaktionsdesign (IDE)* ausleihen, um wieder ein Voiceover aufnehmen zu können. Sie findet das Mikrofon unter *zurückgegeben* und leiht das Material erneut aus.



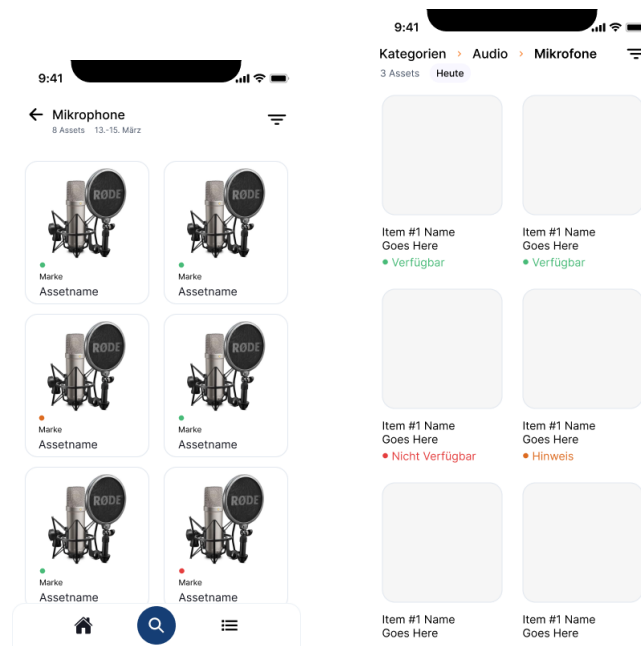


Abbildung 6.4: Kategorien (l), Assetdetails (r)

## 6 Dialogbeispiele

The screenshot shows the Snipe-IT web interface for viewing a specific asset (00087). The interface is divided into a sidebar, a top navigation bar, and a main content area.

**Sidebar (Left):**

- Dashboard
- Assets
  - Alle auflisten
  - Ausleihbar (58)
  - Innerhalb der Uni nutzbar (2)
  - In Wartung (0)
  - Nur am Standort nutzbar (13)
  - Fest verbaut (103)
  - Alle Herausgegeben (103)
  - Alle Bereit zum Herausgeben (59)
  - Alle Ausstehende (0)
  - Alle Nicht herausgebbar (116)
  - Alle Archiviert (0)
  - Anforderbar
  - Audit fällig
  - Audit überfällig
- Massen-Herausgeben
- Angefragt
- Gelöscht
- Asset Wartungen
- Import Verlauf
- Massenprüfung
- Lizenzen
- Zubehör
- Verbrauchsmaterialien
- Komponenten
- Vordefinierte Kits
- Personen
- Import
- Einstellungen
- Berichte
- Anforderbar

**Top Bar:**

- Logo of Universität zu Köln
- Search bar: "Nach Asset Tag suchen"
- Buttons: "Hinzufügen", "Anna Tabara"

**Main Content Area:**

**Asset ansehen 00087**

**Info Tab:**

Info	Lizenzen	Komponenten	Assets	Historie	Wartungen	Dateien
<b>Status</b>	Ausleihbar <a href="#">displayable</a>					
<b>Firma</b>	IDUX					
<b>Asset Name</b>	VR Entwicklungslaptop <CoLab>					
<b>Hersteller</b>	Lenovo <a href="https://www.lenovo.com/de/de">https://www.lenovo.com/de/de</a>					
<b>Kategorie</b>	PC - Laptop					
<b>Modell</b>	Lenovo Legion Creator 7i					
<b>Modellnummer</b>	82HCT01WWDEDE0					
<b>CPU</b>	Intel Core i7-10675H / 2.3 GHz					
<b>GPU</b>	NVIDIA GeForce RTX 2070 Super Max-Q					
<b>RAM</b>	16 GB					
<b>Hauptspeicher</b>	512 GB SSD					
<b>MAC Adresse</b>						
<b>Kaufdatum</b>	01.02.2021					
<b>Kaufpreis</b>	2.163,00					
<b>Lieferant</b>	Amazon					
<b>Notizen</b>						
<b>Standort</b>	Collaboration Lab					
<b>Standard-Standort</b>	Collaboration Lab					
<b>Erstellt am</b>	16.02.2022 09:26					
<b>Aktualisiert am</b>	22.09.2022 09:36					
<b>Herausgaben</b>	4					
<b>Rücknahmen</b>	4					
<b>Anfragen</b>	0					
<b>Labels</b>	<a href="#">Label generieren</a>					

**Asset Image:** A black Lenovo Legion laptop with "LEGION" on the lid. A QR code is visible next to the image.

**Footer:**

Snipe-IT is open source software, made with ❤ by @snipeitapp

Version v5.3.10 - build 6684 (master) [User's Manual](#) [Report a Bug](#)

Abbildung 6.5: Assetansicht in Snipe-IT

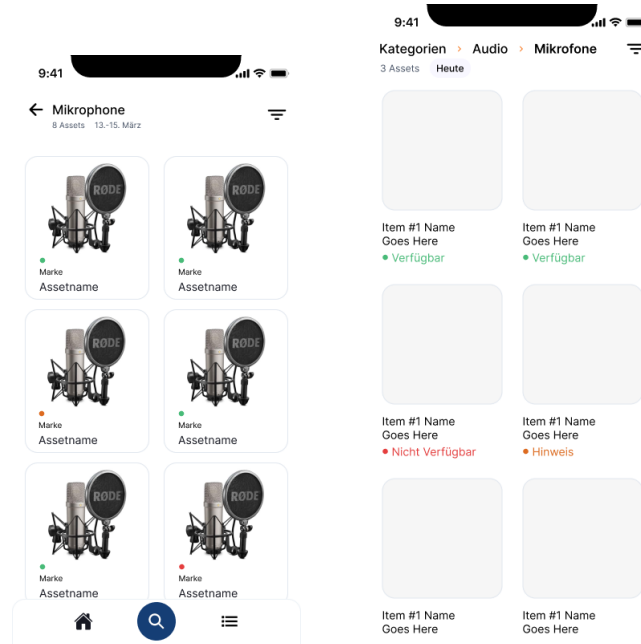


Abbildung 6.6: Kategorien (l), Assetdetails (r)

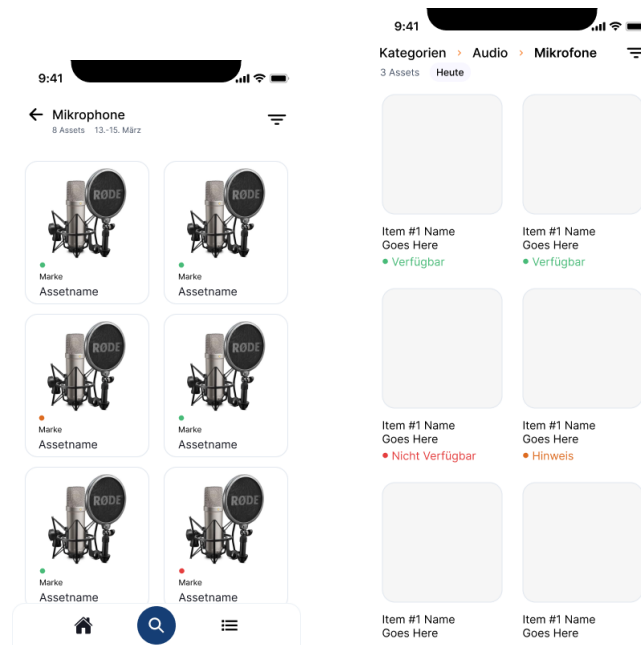


Abbildung 6.7: Kategorien (l), Assetdetails (r)



# 7

## Evaluation

Das folgende Kapitel evaluiert das im Rahmen dieser der Arbeit realisierte System. Zunächst werden die Ziele sowie auf das Vorgehen erläutert. Daraufhin werden die Ergebnisse der Versuchsgruppen, welche sich nach den in der Benutzeranalyse erarbeiteten Gruppen gliedern, betrachtet. Abschließende werden die Ergebnisse entsprechend diskutiert zusammengefasst.

### 7.1 Ziele

Im Rahmen der Forschungsfrage F3 soll herausgearbeitet werden, inwiefern die Gebrauchstauglichkeit und Nützlichkeit durch das in der vorliegenden Arbeit entwickelte Reservierungssystem gewährleistet werden kann. Die Evaluation wird zusätzlich dafür genutzt, das entwickelte Reservierungssystem zu Untersuchung von Forschungsfrage F2 hinsichtlich Funktionen, Gestaltung und Nachvollziehbarkeit bewerten zu lassen.

### 7.2 Vorgehen und Methodik

Formative vs. Summative Evaluations <https://www.nngroup.com/articles/formative-vs-summative-evaluations/>

Da es sich um ein universitätsinternes Tool handelt, wurde sich bei den Versuchspersonen, im Rahmen einer Laborstudie (N=6) mit Mitarbeitenden des IMIS und Studierenden im Bereich der Medieninformatik zusammengesetzt, um das konzipierte System zu evaluieren. Zu Beginn der Studienplanung wurden Evaluationsaufgaben definiert, welche die Versuchspersonen Schritt für Schritt durchführen sollten (Anhang C). Dabei sollten die Teilnehmenden

die Think-Aloud-Methode anwenden, das heißt bei der Nutzung des System sollten Versuchspersonen ständig laut denken und somit ihre Gedanken verbalisieren, während sie sich durch die Benutzeroberfläche bewegen (Nielsen, 1994b). Der Vorteil der Methode ist, dass durch das Beobachten der Nutzenden nicht nur Probleme auffallen, sondern diese auch begründet werden. Zudem können Wünsche oder Erwartungen besser nachvollzogen werden (Nielsen, 2012).

Um die Gebrauchstauglichkeit und Nützlichkeit der Web-App abschließend feststellen zu können, wurde ein Online-Fragebogen entworfen. Zu Beginn des Fragebogens wurden Teilnehmende nach den demografischen Daten gefragt. Diese dienen der besseren Klassifizierung der Daten. Daraufhin wurde die Technikaffinität mithilfe der ATI-Skala erfragt. Im dritten Abschnitt wurden Fragen zu den Funktionen der Anwendung gestellt. Hierbei sollten zum einen bereits vorhandene Funktionen Wichtigkeit zugeordnet werden. Zum anderen sollten noch nicht vorhandene Funktionen angegeben werden. Dies dient unter anderem dem Ausblick und Abwägen der zukünftigen Weiterentwicklung des Systems. Des Weiteren wurde in diesem Teil auf das Wording der Anwendung eingegangen, da diese in der Zwischenevaluation des Interface-Designs vermehrt zu Unverständlichkeiten geführt hat. Abschließend wurde mithilfe des User Experience Questionnaire (UEQ)-Fragebogens die Usability getestet (Burghardt, Wimmer, Wolff & Womser-Hacker, 2018). Schließlich wurden Proband:innen befragt, wie sie das System in seiner Gesamtheit bewerten würden und ob sie sich dieses System für den regelmäßigen Gebrauch vorstellen könnten.

Anhang:  
Umfrage

### 7.3 ATI und UEQ Ergebnisse

Da sich die Anwendung lediglich in Bereich der Verwaltung unterscheidet werden im Folgenden die Technikaffinität sowie Usability der Anwendung beider Versuchsgruppen dargestellt.

Aufgrund von Vollständigkeit sollten die Usability der Anwendung mithilfe des UEQ betrachtet werden. Zu berücksichtigen ist, dass kein Vergleichssystem Vorschlag und die Aussagekraft der Werte fraglich ist?

Werte zwischen -0.8 und 0.8 stehen für eine neutrale Bewertung der entsprechenden Skala, Werte größer als 0.8 für eine positive Bewertung und Werte kleiner als -0.8 für eine negative Bewertung. Der Bereich der Skalen liegt zwischen -3 (sehr schlecht) und +3 (sehr gut).

Die pragmatische Qualität beschreibt die empfundene Fähigkeit eines Systems, welche Nutzen haben, um bestimmten Aufgaben mit dem System zu erfüllen. Pragmatische Qualität als

Verschiedene  
Person  
= ver-  
schiedene  
Präferenzen  
immer  
Relativ

Wahrnehmung von Attributen wie: clear, supporting, useful, controllable Hassenzahl: Utility, Usability (Hassenzahl, 2004).

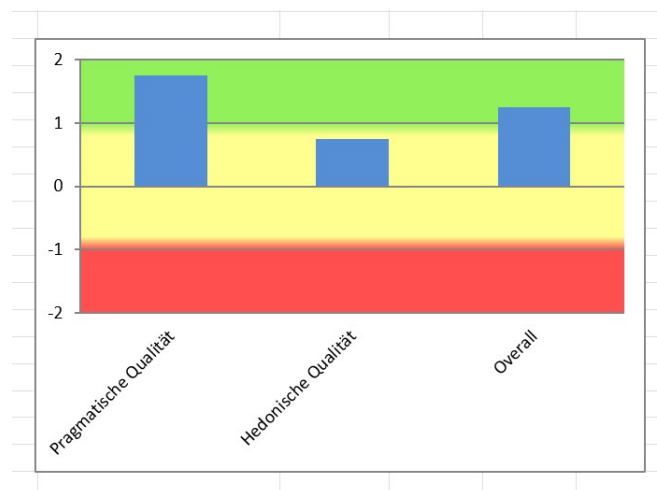
Die hedonische Qualität umfasst die User Experience mit emotionalen und ästhetischen Teilen. Außerdem beschreibt es, was das System symbolisiert oder an neuen Möglichkeiten mitbringt und befriedigt die menschlichen Bedürfnisse nach Neugier und sozialem Vergleich. Hedonische Qualität als Wahrnehmung von Attributen wie: outstanding, impressive, exciting, interesting Kontext: Joy-of-Use, Funology, Pleasure (Hassenzahl, 2004).

**Tabelle 7.1:** Werte der kurzen UEQ-Skala

Pragmatische Qualität	1.75
Hedonische Qualität	0.75
Gesamt	1.25

muss Abb.  
7.1 ? Wenn  
muss  
das noch  
hübsch :D

**Abbildung 7.1:** EQ



**Abbildung 7.2:** Verzeichnis

Des Weiteren wurde mithilfe des ATI das technische Interesse und Verständnis der Teilnehmenden festgestellt (Tabelle 2.3) (Attig et al., 2017). In beiden Gruppen konnten wie zu Beginn in der Literatursection:benutzer lediglich geringe Unterschiede innerhalb der soziodemografischen Daten festgestellt werden.

Insgesamt konnte für beide Gruppen eine Einschätzung der Technikaffinität anhand der ATI-Skala ermittelt werden (Verleihende:  $M=5.XX$ ,  $SD=0.XX$ ,  $N=3$ ; Ausleihende:  $M=5.XX$ ,  $SD=0.XX$ ,  $N=X$ ) (Tabelle 2.3). Durch das Heranziehen zweier Vergleichsstichprobe aus

Franke et al. (2019) ( $M=4.14$ ,  $N=300$  und  $M=4.23$ ,  $N=65$ ), lässt sich schlussfolgern, dass die Nutzendengruppen eine vergleichsweise hohe Technikaffinität aufweist.

**Tabelle 7.2:** Werte der ATI-Skala

Benutzergruppe	Mittelwert ( $M$ )	Standardabweichung ( $SD$ )	Teilnehmende ( $N$ )
Verleihende	5.X	0.X	3
Ausleihende	5.X	0.X	8

## 7.4 Verleihende

Der kommende Abschnitt umfasst lediglich die Evaluationsergebnisse der Aufgaben in Bezug auf den Verwaltungsteil des realisierten Systems, aufseiten der Verleihenden. Tabelle 7.3 zeigt die Rollen der Versuchspersonen. Die IDs der Versuchspersonen werden als Verweise in dem folgenden Abschnitt verwendet.

**Tabelle 7.3:** Teilnehmende der Interviews, Verleihende

ID	Zuständigkeitsbereich
E-V1	Multimedialabor
E-V2	VR-Labor
E-V3	??

Für Verleihende ist es von hoher Bedeutung überprüfen zukönnen, ob die richtige Person das entsprechende Asset abholt. Dies wurde nach dem Klicken auf das Listenelement erwartet (E-V1). Des Weiteren bedarf die Bestätigung, dass ein Asset *abgeholt* oder *zurückgegeben* wurde, bei mehreren Elementen in der Listenansicht, viele Aktionen. Folglich wurde eine Bearbeitungsansicht beim Klicken des Listenelements erwartet oder Vorgeschlagen (E-V1-2).

---

*»›Abgeholt‹, dass muss man bestätigen können. Wenn ich da ausversehen raufklicke [...] wichtig ist eine undo-Oberfläche [...] keine Ahnung wie ich das jetzt zurückhole [...]'«*

---



## 7.5 Ausleihende

Der kommende Abschnitt umfasst die Evaluationsergebnisse der Aufgaben, unter der Anwendung der Think-Aloud-Methode, aufseiten der Ausleihenden. Tabelle 7.5 zeigt die Rolle und das Alter der Versuchspersonen. Die IDs der Versuchspersonen werden als Verweise in dem folgenden Abschnitt verwendet.

**Tabelle 7.5:** Teilnehmende der Evaluation, Ausleihende

ID	Alter	Rolle
E-A1	19 - 25 J.	Masterstudent:in, Hilfswissenschaftler:in
E-A2	19 - 25 J.	Bachelorstudent:in
E-A3	19 - 25 J.	Bachelorstudent:in, Hilfswissenschaftler:in
E-A4	19 - 25 J.	Bachelorstudent:in, Hilfswissenschaftler:in
E-A5	19 - 25 J.	Masterstudent:in

Auf dem Einloggbildschirm wurde der Name der Anwendung erwartet, da die Zuordnung des Tools sonst schwerfallen könnte. Außerdem war die Wortwahl und der Platz zum *Mit IDM Accout einloggen*-Button problematisch, da die Eingabefelder und der Button als zwei unterschiedliche Einlogg-Wege verstanden wurden und nicht als Bestätigungs-Button für das Formular (E-V1, E-A4).

Das Dashboard wurde von den Nutzenden stets als Übersichtlich und hilfreich betitelt, wobei die Namensgebung *Dashboard* als einziges englisches Wort negativ aufgefallen ist (E-V1). In der Beschreibung der Tabs würde ein *deine* förderlich sein, um zu verdeutlichen, dass es sich um die eigens getätigten Reservierungen handelt und nicht um alle Reservierungen (E-A1). Da das Dashboard bei keinen Reservierungen viel freien Platz lässt, wäre eine direkte Übersicht über Kategorien denkbar sinnvoll (E-A3).

Die Kategorien wurden von allen Nutzenden als einfach und wichtig bezeichnet. Positiv wurde angemerkt, dass diese ein schnelles durchsuchen ermöglichen, um zu erfahren, was für Materialien ausgeliehen werden können (E-A3). Ohne Kateogrien sei die Anwendung nur halb so übersichtlich (V-E4).

---

*»[...] , die Kategorien das hätte ich mir nur ein bisschen übersichtlicher gewünscht, [...] und vielleicht auf dem Dashboard schon eine Art Übersicht.«*

---

Eine weitere Beobachtung, in Hinsicht auf die Kategorien, betrifft das aufklappen der Unterkategorien. Nachdem diese geöffnet wurden müssen Nutzende zunächst auf *zurück*-klicken,

um Oberkategorien erneut einsehen zu können (V-E3).

Allen Versuchespersonen ist das Suchen über die Sucheiste sowie die Navigationsleiste leicht gefallen. Ansprechpartner:innen und der Abholort eines Assets wurden auf den ersten Blick entsteckt. Wobei eine Raumnummer bei dem Abholort als Fehlend dekleriert wurde.

---

*»[...] , wenn hier die Kontaktdaten zu den Verantwortlichen sind [...], dass man draufklicken könnte und direkt kontaktieren könnte über die App [...]«*

---

Suche über die Navigationsleiste wurde als modern und ansprechend befunden. Die Suche im Burgermenü wurde nicht von allen Nutzenden als eindeutig und hilfreich betitelt (??). Das Suchen über einen bestimmten Zeitraum erschien sehr praktisch (E-A5). Ergänzend könnte in der Navigationsleiste ein Filtericon eingebaut werden, welche ebenfalls das Suchen über einen Zeitraum beinhaltet (E-A5).

Der Reservierungsprozess wurde durch die Kalenderkomponente zu Teilen erschwärt. Als irritierend galt die fehlende Hervorhebung des aktuellen Tages (E-A1,2). Nicht auswählbare Tage werden bisher lediglich ausgegraut, was die Unterscheidung zwischen generell nicht auswählbaren Tagen (Wochenende) und bereits reservierten Zeiträumen unübersichtlich wirken lässt. Die Kalender und Uhrzeitenansicht wurde bis dato auf Englisch angezeigt, wobei Versuchspersonen anmerkten, dass diese auf Deutsch für die Einheitlichkeit sinnvoller wären (E-A1). Sobald der Zeitraum ausgewählt wurde, erhielten Nutzenden eine Reseriverungsübersicht und eine Zusammenfassungübersicht. Die Bestätigung von zwei Reseriverungsübersichten war für die Versuchsperson unklar. Bereits bei der ersten Übersicht interpretierten Nutzenden die Reservierung als abgeschlossen, einige kehrten zum Dashboard zurück, wo die Reservierung jedoch nicht angezeigt wurde. Dieses Missverständnis wurde unter anderem auf die Betitelung des Butten *weiter* zurückgeführt. Um den Reseriverungsprozess sichtbarer zu gestalten, wurde Vorschlag eine Vortschriftanzeige einzublenden (E-A3). Ein weiterer Punkte, weswegen das Abschließen der Reservierung als (nicht)entgültig interpretiert werden kann, ist das Fehlen eines *Zurück zum Dashboard*-Button auf der Reservierungszusammenfassung (E-A1,3). Des Weiteren hat die Animation der Kalenderkomponente beim Schließen dieser als Bestätigung interpretiert (E-V1, A3).

Um erneut Informationen zum rservierten Material einsehen zu können erschien es als unständlich, dass Nutzenden erneut nach dem Material suchen müssen. Alle Versuchspersonen haben intuitiv auf das im Dashboard angezeigte Listenelement geklickt und eine Verlinkung erwartet (E-V1 bis E-V5, E-A1-2). Eine Versuchsperson hat zunächst auf das Listenelement geklickt, in der Erwartung, dass sich eine Seite zum Bearbeiten des Materials öffnet (E-A4). Beim Bearbeiten des Zeitraums stellte sich das Kalender-Popup, auf der mobile Version, als

unhandlich heraus, da sich automatisch die Tastatur öffnet und die Komponente sich bei danebentippen direkt wieder schloss. Außerdem fehlte eine Bestätigung der Änderung oder ein *Änderungen speichern*-Button (E-A1 bis 5, E-V1-2). Ebenfalls wurde von allen Versuchsperson die im Kalender ausgewählten Tage per ziehen versucht zu verändert.

Das Löschen einer Reservierung hat bei allen Versuchspersonen gut funktioniert und wurde als intuitiv bezeichnet. Zwei Personen wiesen darauf hin, das vor dem Löschen eine Warnung sinnvoll ist, falls Nutzende ausversehen auf Löschen klicken (V-E1. E-A1).

Die Assetstatus *Fest verbaut* und *Am Standort nutzbar* führten bei den Versuchspersonen zu irritation. Versuchspersonen fehlte zu den Betitelung eine Erklärung. Wording oder legende.

---

*» Was bedeutet das für die Reservierung?«*

*»[...] das heißt von ›fest verbaut‹ frage ich mich, ob ich das jetzt trotzdem ausleihen kann oder ob ›fest verbaut‹ heißt, dass ich das nur vor Ort nutzen kann[...]«*

*»[...] das (der Assetstatus) könnte vielleicht ein bisschen salienter sein.«*

---

Generell wurde die Oberfläche als übersichtlich und *clean* beschrieben (E-A1 bis E-A4). Der Wunsch die Anwendung im Universitätalltag für Projekte nutzen zu können wurde entsprechend geäußert (E-A1,3,4).

---

*»Ja, wäre cool, wenn wir das am IMIS wirklich nutzen könnten«*

*»[...] es ist insgesamt sehr übersichtlich und ich würde sagen, dass ich mich auf jeden Fall gut zurechtfinde [...]«*

*»Allgemein die Idee das es wie ein Online-Shop ist [...], weil viele Leute einfach alles neu kaufen, statt es auszuleihen und wenn du den einen Online-Shop an die Hand gibts können die einfach [...] danach suchen.«*

---

Umfrage auswerten lol

## 7.6 Diskussion

# 8

## Zusammenfassung und Ausblick

Die Zusammenfassung greift die in der Einleitung angerissenen Bereiche wieder auf und erläutert, zu welchen Ergebnissen diese Arbeit kommt. Dabei wird insbesondere auf die neuen Erkenntnisse und den Nutzen der Arbeit eingegangen.

Im anschließenden Ausblick werden mögliche nächste Schritte aufgezählt, um die Forschung an diesem Thema weiter voranzubringen. Hier darf man sich nicht scheuen, klar zu benennen, was im Rahmen dieser Arbeit nicht bearbeitet werden konnte und wo noch weitere Arbeit notwendig ist.

Die Resultate der Untersuchungen im Rahmen der Arbeit können als Indikator und Grundlage für weiterführende Projekte und Untersuchungen eingesetzt werden. Insbesondere für Erhebungen mit der Zielgruppe wurden zahlreiche Vorarbeiten geleistet, an die es anzuknüpfen gilt:

### 8.1 Zusammenfassung

### 8.2 Offene Punkte

Funktionalitäten erwähnen!

### 8.3 Ausblick

Weiterentwicklung des Interfacedesigns

Weiterentwicklung der Funktionalitäten

LDAP Einbindung

Einbindung in das Labormanagementsystem

### 8.4 Abschließendes Fazit

Labormanagementsystem

## Abkürzungsverzeichnis

<b>IMIS</b>	Institut für Multimediale und Interaktive Systeme
<b>HTTPS</b>	HyperText Transfer Protocol Secure
<b>ATI</b>	Fragebogen zur interaktionsbezogenen Technikaffinität
<b>HiWi</b>	Hilfswissenschaftler:innen
<b>WiMi</b>	wissenschaftlichen Mitarbeiter:innen
<b>EMI</b>	Einführung in die Medieninformatik
<b>IDE</b>	Interaktionsdesign
<b>DSL</b>	Domain Specific Language
<b>UEQ</b>	User Experience Questionnaire





# Abbildungsverzeichnis

1.1	Vorgehensmodell . . . . .	3
2.1	Anforderungsschablone . . . . .	20
3.1	Architektur Level 1 . . . . .	29
3.2	Architektur Level 2 . . . . .	30
3.3	Architektur Level 3 . . . . .	32
4.1	Typografie von Tailwind UI . . . . .	35
4.2	Farbsystem des Interface-Designs . . . . .	36
4.3	Dashboard in der mobilen Ansicht . . . . .	37
4.4	- . . . .	38
4.5	Materialsuche . . . . .	39
4.6	Materialansicht (Kacheln) . . . . .	40
4.7	Kalenderkomponente . . . . .	41
4.8	Reservierung und Check-out . . . . .	42
5.1	Verzeichnisstruktur des Reservierungsinterfaces . . . . .	44
5.2	Datenstruktur der Reservierungen in Verbindung mit Nutzenden . . . . .	45
5.3	Quellcode: schema.prisma . . . . .	46
5.4	Quellcode: migration.sql . . . . .	47
5.5	UML Snipe-IT API . . . . .	48
5.6	Quellcode: server.ts . . . . .	49
5.7	Verzeichnisstruktur des Clients . . . . .	49
5.8	Quellcode: server.ts . . . . .	50
5.9	Inhalte der .env Datei im Frontend (links) und Backend (rechts) . . . . .	51
6.1	Mockup: Kategorien, Assets, Assetdetails . . . . .	54
6.2	Mockup: Kategorien, Assets, Assetdetails . . . . .	55
6.3	Mockup: Kategorien, Assets, Assetdetails . . . . .	56
6.4	Mockup: Kategorien, Assets, Assetdetails . . . . .	57

6.5	Assetansicht in Snipe-IT . . . . .	58
6.6	Mockup: Kategorien, Assets, Assetdetails . . . . .	59
6.7	Mockup: Kategorien, Assets, Assetdetails . . . . .	59
7.1	EQ . . . . .	63
7.2	UEQ . . . . .	63

## Tabellenverzeichnis

2.1	Teilnehmende der Interviews, Verleihende . . . . .	6
2.2	Teilnehmende der Interviews, Ausleihende (die mit * gekennzeichneten Personen wurden gemeinsam interviewt) . . . . .	7
2.3	Ergebniswerte der ATI-Skala . . . . .	7
2.4	Aufgaben im Bereich der Vorbereitung . . . . .	12
2.5	Aufgaben im Bereich der Ausgabe . . . . .	13
2.6	Aufgaben im Bereich der Rückgabe . . . . .	14
2.7	Aufgaben im Bereich der Wartung . . . . .	15
3.1	Funktionalitäten für VA (Ver- und Ausleihende) . . . . .	24
3.2	Funktionalitäten für (V)erleihenden . . . . .	25
3.3	Funktionalitäten für (A)usleihende . . . . .	27
4.1	Teilnehmende der Zwischenevaluation . . . . .	34
5.1	API des Reservierungsinterfaces . . . . .	44
7.1	Werte der kurzen UEQ-Skala . . . . .	63
7.2	Werte der ATI-Skala . . . . .	64
7.3	Teilnehmende der Interviews, Verleihende . . . . .	64
7.5	Teilnehmende der Evaluation, Ausleihende . . . . .	65



## Quelltextverzeichnis



# Literaturverzeichnis

- Attig, C., Wessel, D. & Franke, T. (2017). Assessing Personality Differences in Human-Technology Interaction: An Overview of Key Self-report Scales to Predict Successful Interaction. In C. Stephanidis (Hrsg.), *HCI International 2017 – Posters’ Extended Abstracts* (S. 19–29). Cham: Springer International Publishing. doi: 10.1007/978-3-319-58750-9\_3
- Balzert, H. (2009). *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering*. Spektrum Akademischer Verlag. Zugriff auf <https://doi.org/10.1007/978-3-8274-2247-7>
- Brown, S. (2013). Software architecture for developers. *Coding the Architecture*.  
*Browser stats: Browser market share*. (o.J.). Zugriff am 2022-05-03 auf <https://www.stetic.com/market-share/browser/>
- Burghardt, M., Wimmer, R., Wolff, C. & Womser-Hacker, C. (2018). *Mensch und Computer 2017 - Tagungsband: Spielend einfach interagieren*. BoD – Books on Demand. (Google-Books-ID: MEJMDwAAQBAJ)
- DIN EN ISO 9421-210:2020-03, *Ergonomie der Mensch-System-Interaktion - Teil 210: Menschzentrierte Gestaltung interaktiver Systeme* (Bd. 2020; DIN). (2020, 3).
- Dresing, T. & Pehl, T. (2016). *Praxisbuch Interview, Transkription & Analyse - Anleitungen und Regelsysteme für qualitativ Forschende*. Marburg: dr. dresing & pehl GmbH.
- Dumke, R. & Lehner, F. (Hrsg.). (2000). *Software-metriken: Entwicklungen, werkzeuge und anwendungsverfahren*. Deutscher Universitätsverlag. Zugriff am 2022-10-04 auf <http://link.springer.com/10.1007/978-3-322-93389-8> doi: 10.1007/978-3-322-93389-8
- Fielding, R. T. & Reschke, J. (2014, Juni). *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content* (Request for Comments Nr. RFC 7231). Internet Engineering Task Force. Zugriff am 2022-10-12 auf <https://datatracker.ietf.org/doc/rfc7231> (Num Pages: 101) doi: 10.17487/RFC7231
- Franke, T., Attig, C. & Wessel, D. (2019, April). A Personal Resource for Technology Interaction: Development and Validation of the Affinity for Technology Interaction (ATI) Scale. *International Journal of Human-Computer Interaction*, 35 (6), 456–467. Zugriff am

- 2022-07-11 auf <https://doi.org/10.1080/10447318.2018.1456150> (Publisher: Taylor & Francis \_eprint: <https://doi.org/10.1080/10447318.2018.1456150>) doi: 10.1080/10447318.2018.1456150
- Google. (2022). *Material Design*. Zugriff am 2022-09-28 auf <https://material.io/components/navigation-drawer#specs>
- Gordon, K. (2021). *Using Color to Enhance Your Design*. Zugriff am 2022-09-27 auf <https://www.nngroup.com/articles/color-enhance-design/>
- Hassenzahl, M. (2004). The Thing and I: Understanding the Relationship Between User and Product. In M. A. Blythe, K. Overbeeke, A. F. Monk & P. C. Wright (Hrsg.), *Funology: From Usability to Enjoyment* (S. 31–42). Dordrecht: Springer Netherlands. Zugriff am 2022-10-26 auf [https://doi.org/10.1007/1-4020-2967-5\\_4](https://doi.org/10.1007/1-4020-2967-5_4) doi: 10.1007/1-4020-2967-5\_4
- Herczeg, M. (2009a). *Einführung in die medieninformatik*. Oldenbourg Wissenschaftsverlag. Zugriff auf <https://www.degruyter.com/document/doi/10.1524/9783486593464/html?lang=de> (Einführung in die Medieninformatik) doi: 10.1524/9783486593464
- Herczeg, M. (2009b). *Interaktionsdesign*. München: Oldenbourg Wissenschaftsverlag. Zugriff am 2022-10-04 auf <https://doi.org/10.1524/9783486594942> doi: 10.1524/9783486594942
- Herczeg, M. (2018). *Software-Ergonomie: Theorien, Modelle und Kriterien für gebrauchstaugliche interaktive Computersysteme* (4. Auflage Aufl.). Berlin ; Boston: De Gruyter Oldenbourg. (Backup Publisher: Walter de Gruyter GmbH & Co. KG)
- Kim, B. (2013, September). Chapter 4: Responsive Web Design, Discoverability, and Mobile Challenge. *Library Technology Reports*, 49 (6), 29–39. Zugriff am 2022-09-27 auf <https://www.journals.ala.org/index.php/ltr/article/view/4507> (Number: 6)
- leserlich.info. (2022). *leserlich.info – Schritte zu einem inklusiven Kommunikationsdesign*. Zugriff am 2022-10-03 auf <https://www.leserlich.info>
- Nielsen, J. (1994a). Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (S. 152–158). Association for Computing Machinery. Zugriff am 2022-09-27 auf <https://doi.org/10.1145/191666.191729> doi: 10.1145/191666.191729
- Nielsen, J. (1994b). *Usability engineering*. Morgan Kaufmann. (Google-Books-ID: 95As2OF67f0C)
- Nielsen, J. (2012). *Thinking aloud: The #1 usability tool*. Zugriff am 2022-10-19 auf <https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/>
- Pohl, K., Lauenroth, K., Sikora, E., Weyer, T., Froese, A., Halmans, G. & Metzger, A.



- (2008). *Requirements Engineering: Grundlagen, Prinzipien, Techniken* (2., korr Aufl.). Heidelberg: dpunkt.Verlag GmbH.
- Snipe-IT. (o. J.). *Home - snipe-IT open source IT asset management*. Zugriff auf <https://snipeitapp.com>
- Söderholm, J. (2018). *Borrowing and lending tools: The materiality of x-lending libraries* (Unveröffentlichte Dissertation). Höskolan i Borås.
- ZHB, m. (o. J.). *Öffnung unter Corona-Bedingungen*. Zugriff am 2022-04-19 auf [https://www.zhb.uni-luebeck.de/index.php?id=20&tx\\_news\\_pi1%5Bnews%5D=72&tx\\_news\\_pi1%5Bcontroller%5D=News&tx\\_news\\_pi1%5Baction%5D=detail&cHash=259293382214e092f4d692df71e59c53](https://www.zhb.uni-luebeck.de/index.php?id=20&tx_news_pi1%5Bnews%5D=72&tx_news_pi1%5Bcontroller%5D=News&tx_news_pi1%5Baction%5D=detail&cHash=259293382214e092f4d692df71e59c53)



# Anhang

Im Folgenden werden ...

## A Interviewleitfaden der Analyse

- Begrüßung und Danken für die Zeit
- Kurzer Umriss des Themas: Ziel in den Interviews ist es, zu erfahren, wie die Organisation und Planung unter den Mitarbeiter:innen untereinander sowie zwischen den Mitarbeiter:innen und Studierenden aktuell abläuft und wie diese Kommunikation verbessert werden kann, in Bezug auf das Ausleihen von Assets + in 2 Teile geteilt
- Tätigkeit
- Datenschutz
- Wenn einverstanden sind, Interview aufzeichnen und würden Sie bitten, dazu das Datenschutzformular auszufüllen. Die Aufzeichnungen werden diskret behandelt und nach der Auswertung gelöscht

### A.1 Verleihende

#### Abschnitt: Jetzt

1. Interessieren, wie Sie vorgehen, wenn Sie eine Anfrage erhalten zum Ausleihen eines Assets.
  - a) Bekommen Sie anfragen per Mail
  - b) oder mündlich
  - c) Kalender Eintragen (Outlook)/Zettel
  - d) Einfach nehmen

2. Gibt es eine öffentliche Übersicht (für Studierende) der auszuleihenden Geräte? Nein:  
Hat das schon einmal für Probleme gesorgt im Ausleihprozess? Im Nachhinein gehört,  
dass Assets benötigt wurden
3. Können Personen vorläufig Systeme reservieren, z. B. in 2 Wochen für 4 Tage? Warum  
nicht? Probleme
4. Wird sichergestellt, dass Ausleihende mit dem Gerät umgehen können?
5. Wie sind die Assets versichert bzw. was passiert, wenn es kaputtgeht?
6. Sehen Sie Probleme oder sind Sie mit dem derzeitigen Ablauf zufrieden?
7. Fehlgeleitet Anfragen, direkte Ansprechpartner:innen

#### Abschnitt: Visionen und Ziele

8. Vorstellen, es gibt eine webbasierte Anwendung (System)- Online-Plattform
  - a) Übersicht – Vorstellung? Informationen werden benötigt Form der Darstellung
  - b) Was wären weitere Funktionen?
  - c) Vorausplanen? Reservieren?
  - d) Fehlgeleitet Anfragen, direkte Ansprechpartner:innen
9. Sind Ihnen bis hierher noch Gedanken gekommen, die Sie gerne mit auf den Weg geben  
wollen?
10. Vielen Dank, dass Sie uns für dieses Interview zur Verfügung gestanden haben, wir  
wären jetzt am Ende des Interviews angelangt
11. Im Nachhinein noch melden können

## A.2 Ausleihende

- Begrüßung und Danken für die Zeit
- Kurzer Umriss des Themas
- Vorerfahrung (HiWi,...)
- Datenschutz

### Abschnitt: Jetzt

1. Ist Dir/Ihnen bekannt, welche Assets sie/du am IMIS ausleihen können?
  - a) Nein, was hätten sie/du dann gebraucht? Jetzt, wo du es weißt, würdest du es gerne nutzen? oder Listenübersicht
  - b) Vorgehen, wenn Sie eine Anfrage stellen zum Ausleihen eines Assets.
    - Schauen Sie auch spontan in den Laboren, nach den Geräten vorbei?
    - Wie häufig ist Ihr spontaner Besuch (nicht) erfolgreich?
    - Was machen Sie, wenn Sie keine Person antreffen, der sie mitteilen, dass sie das Gerät mitnehmen?
    - Was wäre für Sie der einfachste Weg, Informationen zu hinterlassen, was würden Sie sich wünschen?
  - c) Sehen Sie hier Probleme oder sind Sie mit dem derzeitigen Ablauf zufrieden?
    - Darstellung von auszuleihenden Inhalten hilfreich (Übersicht)
    - Wie gehen sie vor, wenn sie etwas in 2 Wochen ausleihen wollen?

### Abschnitt: Visionen und Ziele

8. Vorstellen, es gibt eine webbasierte Anwendung (System)- Online-Plattform
  - a) Übersicht – Vorstellung? Informationen werden benötigt Form der Darstellung
  - b) Was wären weitere Funktionen?
  - c) Vorausplanen? Reservieren?
  - d) Fehlgeleitet Anfragen, direkte Ansprechpartner:innen

9. Sind Ihnen bis hierher noch Gedanken gekommen, die Sie gerne mit auf den Weg geben wollen?
10. Vielen Dank, dass Sie uns für dieses Interview zur Verfügung gestanden haben, wir wären jetzt am Ende des Interviews angelangt
11. Im Nachhinein noch melden können

## B Die zehn Usability Heuristiken nach Jakob Nielsen

1. **Sichtbarkeit des Systemstatus** Das System informiert den Nutzer immer darüber, was gerade passiert – rechtzeitig und durch angemessenes Feedback.
2. **Übereinstimmung von System und Wirklichkeit** Das System spricht die Sprache des Nutzers – mit ihm vertrauten Wörtern, Phrasen und Konzepten. Entlehnt aus der echten Welt erscheinen Informationen in ihrer natürlichen und logischen Ordnung.
3. **Nutzerkontrolle und Freiheit** Nutzer führen Aktionen oft unbeabsichtigt durch. Auswege wie „Rückgängig“, „Wiederholen“ und „ESC“ sind deshalb immer möglich und sichtbar.
4. **Beständigkeit und Standards** Nutzer müssen nicht überlegen, ob unterschiedliche Wörter, Situationen und Aktionen das Gleiche meinen. Die Konventionen des Betriebssystems werden eingehalten.
5. **Fehlervermeidung** Besser als jede gute Fehlermeldung ist ein sorgfältiges Design, welches Fehler gar nicht erst auftreten lässt. Das System vermeidet fehleranfällige Situationen oder warnt den Nutzer und lässt ihn die Aktion bestätigen.
6. **Wiedererkennung statt Erinnerung** Durch sichtbare Objekte, Aktionen und Optionen muss der Nutzer weniger im Gedächtnis behalten. Anleitungen zum Gebrauch des Systems sind sichtbar oder leicht zu erreichen.
7. **Flexibilität und Effizienz** Kurzbefehle und andere Abkürzungen – unsichtbar für Neulinge – beschleunigen bei fortgeschrittenen Nutzern die Bedienung. Zusätzlich sind häufige Aktionen individuell anpassbar.
8. **Ästhetisches und minimalistisches Design** Dialogfenster enthalten keine überflüssigen oder nur selten gebrauchten Informationen. Denn jede zusätzliche Information steht in Konkurrenz mit den relevanten Informationen und mindert deren Sichtbarkeit.

9. **Hilfestellung beim Erkennen, Bewerten und Beheben von Fehlern** Fehlermeldungen sollten in klarer Sprache (kein Code) formuliert sein, das Problem exakt beschreiben und eine konstruktive Lösung vorschlagen.
10. **Hilfe und Dokumentation** Obwohl es besser ist, wenn der Nutzer ein System ohne Hilfe benutzen kann, ist es manchmal Nötig, eine Dokumentation bereitzustellen. In dem Fall sind die Informationen einfach zu finden und konzentrieren sich auf die Aufgabe des Nutzers. Die Dokumentation enthält konkrete Schritte zur Ausführung und beschränkt sich auf das Wesentliche.

## C Leitfaden und Aufgaben der Evaluation

1. Begrüßung
2. Erklärung des Evaluationsablauf (Nennung von Erhebungsmethoden, Rahmenbedingungen, Datenschutz)
3. Zeit für Fragen zu dem Evaluationsablauf
4. Einleitung
5. Einführung in den Rahmen der Arbeit/des Projektes
6. Zeit für Fragen zu der Einführung
7. Erklärung, Ziel und Ablauf der Evaluationsaufgaben (Erläuterung Think-Aloud)
8. Zeit für Fragen zu den Evaluationsaufgaben (Zeit dafür, dass Teilnehmer:in sich die Fragen kurz durchlesen kann)
9. Durchführung der Evaluationsaufgaben (Anwendung Think-Aloud Methode, Beobachtung, Notizen)
10. Zeit für Fragen zu dem Fragebogen
11. Fragebogen
12. Am Ende Bedanken für die Teilnahme

### C.1 Verleihende

**Szenario:** Mitarbeitende und möchtest überprüfen, wann Abholungen anstehen und ob du Material für eine Studie gebrauchen kannst

1. Einloggen in die App
  - voraussichtliche Aufgabendauer: 20 Sekunden
  - Obergrenze: 1:30 Minuten
2. Entwicklungslaptop wurde bereits abgeholt
  - voraussichtliche Aufgabendauer: 2 Minuten
  - Obergrenze: 3:30 Minuten
3. Finde das Gerät XY und bringe in erfahre, wo du das Gerät nutzen kannst
  - voraussichtliche Aufgabendauer: 1 Minute
  - Obergrenze: 1:30 Minuten
4. Zeitsprung: Laptop zurückgebracht
  - voraussichtliche Aufgabendauer: 1 Minute
  - Obergrenze: 2:00 Minuten
5. Entwicklungslaptop im Zeitraum 24. Oktober-27. Oktober ausleihen (Uhrzeit wie es in den Alltag passt)
  - voraussichtliche Aufgabendauer: 30 Sekunden
  - Obergrenze: 1:30 Minuten
6. Zeit zum Umschauen geben



## C.2 Ausleihende

Szenario: Studierende und möchtest für ein Projekt Material ausleihen

1. Einloggen in die App
  - voraussichtliche Aufgabendauer: 20 Sekunden
  - Obergrenze: 1:30 Minuten
2. Entwicklungslaptop im Zeitraum 24. Oktober-27. Oktober ausleihen (Uhrzeit wie es in den Alltag passt)
  - voraussichtliche Aufgabendauer: 2 Minuten
  - Obergrenze: 3:30 Minuten
3. Informationen zur Abholung des Laptops (Kontaktdaten, Ort)
  - voraussichtliche Aufgabendauer: 20 Sekunden
  - Obergrenze: 1:30 Minuten
    - Frage zu dem Laptop
4. Material: Gurte Status herausfinden
  - voraussichtliche Aufgabendauer: 1 Minute
  - Obergrenze: 1:30 Minuten
5. Zeitsprung: Laptop abgeholt: Reservierungszeitraum des Laptops verlängern
  - voraussichtliche Aufgabendauer: 1 Minute
  - Obergrenze: 2:00 Minuten
6. Webcam Reservierung löschen
  - voraussichtliche Aufgabendauer: 30 Sekunden
  - Obergrenze: 1:30 Minuten
7. Zeitsprung: Laptop erneut ausleihen
  - voraussichtliche Aufgabendauer: 30 Sekunden
  - Obergrenze: 1:30 Minuten

8. Zeit zum Umschauen geben

## D Umfrage der Evaluation

### D.1 Ausleihende

### D.2 Verleihende

## E Digitale Medien

Der vorliegenden Arbeit ist ein digitaler Anhang auf einer CD beigelegt. Darauf befinden sich:

1. Transkripte der geführten Interviews (Unterverzeichnis Befragungen)
2. Ergebnisse des Reichweiten-Workshops (Unterverzeichnis Workshop)
3. Detaillierte Ergebnisse der Evaluation (Unterverzeichnis Evaluation)
4. Die finale Version der Software (Unterverzeichnis Software)
5. Eine digitale Ausgabe der Arbeit als PDF-Datei (Hauptverzeichnis)

## Erklärung

Ich erkläre hiermit an Eides statt, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

---

(Anna-Tabea Manske)

Lübeck, den 03. Novemeber 2022