

Contents

1 Übersicht	1
2 Einführung	1
3 Anforderungen	1
3.1 Funktionale Anforderungen an die Web-Applikation	2
3.2 Nicht-funktionale Anforderungen an die Web-Applikation	3
3.3 Zusätzliche Anforderungen	3
4 Tech Stack	4
4.1 Framework	4
4.2 Programmiersprache	4
4.3 Datenbank	5
5 Arbeitsprozess	6
5.1 Entwicklungsmodell	6
5.2 Entwicklungsumgebung	6
5.3 Versionskontrolle	6
6 Umsetzung	7
6.1 Anwendungsorientierte Sichtweise	7
6.2 Technikorientierte Sichtweise	17
7 Diskussion	17
8 Ausblick	17

1 Übersicht

2 Einführung

Die Verwaltung von Messmitteln am Fachgebiet Fahrzeugantriebe der Technischen Universität Berlin gestaltet sich schwierig. Zurzeit werden Messmittel dezentral verwaltet. Die handelnden Personen besitzen wenig Information über den Ausleihzustand einzelner Messmittel über verschiedene Projekte hinweg. Daher hat sich das Fachgebiet entschlossen, im Rahmen des Moduls “Projekt Fahrzeugantriebe” eine Web-Applikation zur Verwaltung von Messmitteln zu erstellen. Diese Web-Applikation soll die Ausleihe und Rückgabe einzelner Messmittel über einen QR-Code realisieren. Darüber hinaus sollen Messmittel gruppiert, zusätzliche Informationen, wie zum Beispiel Datenblätter für Messmittel, bereitgestellt und der Bestand von Messmitteln erfasst werden. Um die unterschiedlichen Verantwortlichkeiten der Akteure zu berücksichtigen, ist zudem eine rechtebasierte Nutzerverwaltung vorzusehen.

Der Inhalt dieses Berichtes umfasst die Dokumentation des Arbeitsprozesses, die Beschreibung des finalen Produktes hinsichtlich technischer Umsetzung und Funktionalität und zeigt zusätzlich Anknüpfungspunkte für Folgeprojekte auf.

Messmittel = Sensor

3 Anforderungen

Aus der vom Fachgebiet bereitgestellten Aufgabenstellung wurden unmittelbar die folgenden Anforderungen abgeleitet: Aufbauend auf einer Literaturrecherche soll ein Konzept zur Erstellung einer Web-Applikation entworfen werden. Die Umsetzung besteht aus der Programmierung der entworfenen Web-Applikation sowie dem Anlegen einer zugehörigen Datenbank auf dem Server des Fachgebiets. Die verwendeten Technologien

sind dabei begründet frei zu wählen. Die konkreten Anforderungen an die Web-Applikation sind dabei aus dem in der Aufgabenstellung beschriebenen Funktionstest abgeleitet worden:

- Erstellen von Sensoren
- Erstellen von Nutzern
- Matching unterschiedlicher Sensortypen mit jeweils einem QR-Code
- Ausleihe und Rückgabe von Sensoren
- Konsistente Datenverwaltung in Form einer Datenbank

Diese Anforderungen stellen den minimalen Satz an Anforderungen dar. Während der Bearbeitung des Projektes zeigte sich, dass zusätzliche Anforderungen notwendig sind, um einen zweckmäßigen Einsatz am Fachgebiet sicherzustellen. Die Web-Applikation soll beispielsweise sowohl von Studenten, wissenschaftlichen Mitarbeitern und speziell geschultem Personal (üblicherweise wissenschaftliche Mitarbeiter) mit unterschiedlichen Verantwortlichkeiten genutzt werden. Daher müssen die Minimalanforderungen erweitert werden. Nutzer besitzen unterschiedliche Rechte und müssen sich gegenüber der Web-Applikation authentifizieren. Der finale Satz an Anforderungen kann den folgenden Abschnitten zu funktionalen und nicht-funktionalen Anforderungen entnommen werden.

3.1 Funktionale Anforderungen an die Web-Applikation

Funktionale Anforderungen spiegeln den Funktionsumfang der Web-Applikation wider. Die Vielzahl funktionaler Anforderungen bedingt ein Aufgliedern in verschiedene Funktionsbereiche, die im Folgenden detailliert beschrieben werden.

3.1.1 Sensorverwaltung

Die Sensoren am Fachgebiet stehen untereinander in Beziehung: Ein Versuchsaufbau kann sich aus verschiedenen Sensoren zusammensetzen. Dabei ist weniger der Sensor als viel mehr seine Art von Interesse. So ist es zum Beispiel unerheblich von welchem Hersteller oder aus welcher Serie ein konkreter Sensor ist, solange die gleiche Funktionalität erbracht wird. Der konkrete Sensor wird als *Untertyp* bezeichnet. Die Funktionalität, die alle *Untertypen* verbindet, spiegelt sich im *Typen* wider. Ein Versuchsaufbau ist schließlich eine *Gruppe*, die verschiedene *Typen* beinhaltet. Zusammenfassend lässt sich festhalten:

Jeder Sensor hat einen *Untertypen*. Mehrere nur geringfügig unterschiedliche *Untertypen* werden in einem *Typen* zusammengefasst. Mehrere *Typen* können Teil einer *Gruppe* sein. Ein *Typ* kann Teil mehrerer Gruppen sein.

Ausgehend von diesen Definitionen sind die folgenden Anforderungen zu erfüllen:

- Erstellen und Entfernen von *Gruppen*, *Typen* und *Untertypen*
- Umbenennen von *Gruppen*, *Typen* und *Untertypen*
- Bestandserfassung auf Ebene der *Untertypen*
- Matching von QR-Codes auf Ebene der *Typen*
- Verknüpfung von *Gruppen* mit *Typen* und von *Typen* mit *Untertypen*

3.1.2 Dateiverwaltung

- Hochladen, Umbenennen und Löschen von Datenblättern im PDF-Format für *Gruppen*, *Typen* und *Untertypen*
- Herunterladen von einzelnen Datenblättern als PDF-Datei oder mehreren Datenblättern komprimiert in einer zip-Datei

3.1.3 Nutzerverwaltung

Wie eingangs beschrieben, ist eine rechtebasierte Verwaltung notwendig. Dazu werden drei Rollen angelegt. Der *Benutzer* kann Sensoren ausleihen und zurückgeben. Er kann seinen Benutzernamen und sein Passwort ändern. Der *Moderator* kann zusätzlich *Benutzer* anlegen. Der *Administrator* kann zusätzlich auf die

Sensorverwaltung zugreifen und Nutzer mit einer beliebigen Rolle anlegen. Eine genaue Aufschlüsselung der Rechte der drei Rollen - und somit der Anforderungen an die Nutzerverwaltung - kann Abbildung ... entnommen werden.

TODO: Rechte-Matrix

3.1.4 Ausleihverwaltung

- Ausleihe und Rückgabe von *Untertypen* beliebiger Menge innerhalb eines verfügbaren Rahmens durch Nutzer oder für beliebigen Nutzer durch *Administrator*
- Abschreiben von *Untertypen* durch *Administratoren*
- Anzeige von Datenblättern für *Gruppen*, *Typen* und *Untertypen*

3.1.5 Bestandsinformation

Um ohne Betreten des Lagerortes ermitteln zu können, wie viele Elemente eines *Typen* oder *Untertypen* verfügbar sind oder festzustellen, welcher Nutzer einen benötigten *Typen* ausgeliehen hat, ist eine Übersicht über den Bestand und die Ausleihhistorie zu implementieren.

3.2 Nicht-funktionale Anforderungen an die Web-Applikation

Für einen nachhaltigen Einsatz der Web-Applikation sind die folgenden nicht-funktionalen Anforderungen zu erfüllen:

- Intuitive Nutzerführung
- Konsistenz durch wiedererkennbares Layout und Design
- Performance

Aufgrund ihres nicht-funktionalen Charakters ist die Erfüllung nicht an konkrete Bedingungen geknüpft. Alle unternommenen Bestrebungen zur Erfüllung der funktionalen Anforderungen sind stets hinsichtlich der hier aufgeführten nicht-funktionalen Anforderungen zu bewerten.

3.3 Zusätzliche Anforderungen

Zusätzlich soll eine Datenbank für die konsistente Verwaltung der anzulegenden Daten genutzt werden. Die Web-Applikation und die Datenbank sollen auf einer virtuellen Maschine (*VM*), die auf einem Server des Fachgebiets abgelegt wird, betrieben werden.

3.3.1 Datenbank

Die Datenbank soll die konsistente Datenverwaltung bewerkstelligen. Sie enthält Tabellen, welche entsprechend der funktionalen Anforderungen der Web-Applikation zu gestalten und miteinander zu verknüpfen sind. Hierzu müssen sowohl ein geeignetes Datenbankmodell als auch ein konkretes Datenbankmanagementsystem ausgewählt werden.

3.3.2 Deployment

Als Deployment wird die Integration der Web-Applikation und der Datenbank in die bestehende Infrastruktur bezeichnet. Dazu sind folgende Schritte notwendig:

- Auswahl einer Virtualisierungssoftware
- Einrichten einer *VM*
- Installation von Servern, Wartungssoftware und Programmiersprache
- Transfer von Datenbank und Web-Applikation auf *VM*

4 Tech Stack

Als Tech Stack wird die Summe der verwendeten Technologien bezeichnet. Dazu gehören zum Beispiel die Programmiersprache, das Framework für die Web-Applikation, die Datenbanksoftware, aber auch die Entwicklungsumgebung und weitere Software, die im Entwicklungsprozess verwendet wird.

4.1 Framework

Zuallererst muss das Framework zur Erstellung der Web-Applikation gewählt werden. Dieses legt normalerweise die zu verwendende Programmiersprache fest und setzt möglicherweise Restriktionen in Bezug auf weitere Software. Eine Web-Applikation zeichnet sich dadurch aus, dass sie im Webbrowser ausführbar ist. Der Webbrowser ist in der Lage, Dateien im HTML-Format (HTML: Hyper Text Markup Language) darzustellen. Das HTML-Format spezifiziert dabei ausschließlich die Struktur der Webseite. Um die visuelle Erscheinung der Webseite zu beeinflussen, können Regeln in CSS-Dateien (CSS: Cascading Style Sheet) hinterlegt werden. Für interaktives Verhalten existiert die Sprache JavaScript, die es ermöglicht, das HTML-Dokument dynamisch anzupassen.

Ein Framework zur Erstellung von Web-Applikationen bietet ein Grundgerüst für Layout sowie Funktionalität und stellt einen Server bereit. Für das Layout werden beispielsweise makroskopische Komponenten, wie Dashboards und Landing Pages, oder mikroskopische Komponenten, wie Inputs, Tabellen und Plots, bereitgestellt. Die Funktionalität wird abstrahiert und der Zustand der Web-Applikation modelliert. Der Server bearbeitet Anfragen von Clients, also Nutzern der Web-Applikation. Frameworks können in beliebigen Programmiersprachen implementiert werden, solange eine Schnittstelle zwischen der vom Framework verwendeten Sprache und einer dem Browser verständlichen Sprache existiert. Frameworks können hinsichtlich verschiedener Kriterien unterschieden werden. Backend-Frameworks integrieren neben einem Server meist auch noch Datenbanken, wohingegen Frontend-Frameworks ihren Fokus mehr auf dem visuellen Part legen. Multipage-Frameworks enthalten mehrere Seiten, wohingegen Singlepage-Frameworks nur eine einzige Seite darstellen.

Für die Bearbeitung dieses Projektes wurde das Framework Shiny gewählt, das in der Programmiersprache R implementiert ist. Hierbei handelt es sich um ein Singlepage-Framework, das als Backend den sogenannten Shiny Server enthält. Maßgeblich für die Entscheidung war, dass die beiden Autoren über Erfahrung im Umgang mit R und im Speziellen mit Shiny verfügen. Darüber hinaus zeichnet sich Shiny durch folgende Eigenschaften und Vorzüge aus:

- Moderne Templates
- Fokus auf konkrete Funktionalität
- Reaktives Zustandsmodell
- Modularisierbarkeit
- Für Anwendungsfall ausreichende Performance
- Einfache Integration von Datenbanken

4.2 Programmiersprache

R ist eine Multiparadigmen-Programmiersprache. Je nach Anwendungsfall kann somit zum Beispiel objektorientiert oder funktional programmiert werden. R verfügt einen Pool an Standardbibliotheken und kann einfach durch selbstgeschriebene und frei verfügbare Packages erweitert werden. Das Comprehensive R Archive Network (CRAN) stellt eine Vielzahl von quelloffenen Bibliotheken zur Verfügung beispielsweise {shiny}, das die Funktionalitäten des Frameworks beinhaltet. Die folgenden Tabellen geben Aufschluss über die im Projekt verwendeten Packages und ihren Zweck. Packages können von anderen Packages abhängen. Es wird daher darauf verzichtet auf untergeordnete Bibliotheken einzugehen.

4.2.1 Packages für {shiny}

Package	Beschreibung
{bs4Dash}	AdminLTE-Template
{DT}	DataTables für {shiny}
{htmltools}	HTML-Repräsentation in R
{rclipboard}	Zwischenablage
{shinydisconnect}	Verbindungsverlustbildschirm
{shinyjs}	Integration von Custom-JavaScript
{waiter}	Ladebildschirm

4.2.2 Packages für die Programmierung

Package	Beschreibung
{Cairo}	PDF-/PNG-/SVG-Erstellung
{DBI}	Datenbankinterface
{dplyr}	Datentransformationen
{glue}	String-Erzeugung
{lubridate}	Datumsformat
{RSQLite}	SQLite-Datenbank
{stringr}	String-Manipulation
{tibble}	Tabellenformat
{purrr}	Funktionale Programmierung
{qrcode}	Erstellung von QR-Codes
{renv}	Packagemanagement
{yaml}	YAML-Dateiformat

4.3 Datenbank

Zur konsistenten Datenverwaltung wird eine Datenbank benötigt. Datenbanken sind in der Lage, Anfragen von verschiedenen Clients zu bearbeiten und dabei zu gewährleisten, dass bestimmte Regeln hinsichtlich der Datenstruktur und Ausprägung der Daten eingehalten werden. Es existieren verschiedene Datenbankmodelle, unter anderem das Netzwerk-, das objektorientierte, das hierarchische oder das relationale Datenbankmodell.¹ Diese unterscheiden sich hinsichtlich der Verknüpfung der beteiligten Daten. Aufgrund der hohen Flexibilität und der weiten Verbreitung wurde das relationale Datenbankmodell ausgewählt. Dieses speichert die Daten in miteinander verknüpften Tabellen. Die Tabellenzeilen enthalten Beobachtungen, die Tabellenspalten stellen die beobachtbaren Merkmale dar. Zur eindeutigen Identifikation erhält jede Zeile eine Identifikationsnummer. Die Spalte der Identifikationsnummern wird als Primärschlüssel (*Primary Key*) bezeichnet. Um verschiedene Tabellen miteinander zu verknüpfen, werden Identifikationsnummern referenziert. Eine Spalte, die auf einen Primärschlüssel einer anderen Tabelle verweist, wird als Fremdschlüssel (*Foreign Key*) bezeichnet.

Es gibt eine Vielzahl verschiedener relationaler Datenbankmanagementsysteme, die sich hinsichtlich ihrer Anwendungsbereiche und Skalierbarkeit unterscheiden. Für Projekte kleinen und mittleren Umfangs (unter 100.000 Aufrufe / Tag²) eignet sich SQLite. Hierbei werden alle Tabellen in einer einzigen Datei mit dem Suffix *.sqlite* gespeichert. Der Zugriff auf die Datenbank erfolgt grundsätzlich über die *Structured Query Language* (SQL). Für die Programmiersprache R gibt es die Packages {DBI} und {RSQLite}, die eine direkte Schnittstelle zur Datenbank bereitstellen.

¹Laudon, K.C.; Laudon, J.P.; Schroeder, D.: *Wirtschaftsinformatik - Eine Einführung*, Pearson, 2015, S. 295-300

²SQLite Consortium: *Appropriate Uses for SQLite*, URL: <https://www.sqlite.org/whentouse.html>, Letzter Aufruf: 14.03.2021

5 Arbeitsprozess

Neben der technischen Umsetzung ist insbesondere der Arbeitsprozess von herausragender Bedeutung. Durch diesen wird festgelegt, in welcher Weise die Anforderungen letztendlich umgesetzt werden. Für Projekte im Allgemeinen beinhaltet der Arbeitsprozess das Zeit-, Personal- und Aufgabenmanagement. Ein gut strukturierter Arbeitsprozess hilft schließlich dabei, Ergebnisse effizient und nachvollziehbar zu erzielen. In der Softwareentwicklung wird der Arbeitsprozess zusätzlich durch die Wahl von Entwicklungsmodell, Entwicklungsumgebung und Versionskontrolle bestimmt.

5.1 Entwicklungsmodell

In der Softwareentwicklung existiert eine Vielzahl verschiedener Modelle zur Bewältigung eines Projektes. Die klassischen Modelle (Wasserfallmodell, V-Modell) fokussieren sich darauf, Phasen sequentiell abzuarbeiten. Im Kontrast dazu stehen die agilen Modelle, in denen alle Phasen wiederholt durchlaufen werden. Agile Modelle sind in der Regel deutlich flexibler, da die Anforderungen kontinuierlich angepasst werden können. Für kleine Entwicklungsteams bietet sich die Verwendung eines agilen Entwicklungsmodells auch deswegen an, weil der Koordinationsaufwand zwischen den Teilnehmern gering ist. Das Entwicklungsmodell legt nicht nur fest, wie das Projekt auf der Makroebene strukturiert ist, sondern auch wie auf der Mikroebene konkret programmiert wird.

Wir haben uns dafür entschieden, das agile Modell des Pair Programming - einer Unterform des Extreme Programming - einzusetzen. Hierbei arbeiten stets zwei Programmierer (also im vorliegenden Fall alle) gemeinsam an der Erstellung von Programmcode. Vorteile hiervon liegen im stetigen Informationsaustausch, gemeinsamer Entscheidungsfindung und geringerer Fehlerhäufigkeit. Als nachteilig wird allgemein der doppelte Personalaufwand angesehen.

5.2 Entwicklungsumgebung

Als Entwicklungsumgebung wird die Software bezeichnet, die zur Erstellung und Verwaltung des Programmcodes genutzt wird. Für die Programmiersprache R empfiehlt es sich, die Entwicklungsumgebung RStudio zu verwenden. Diese ermöglicht es, Projekte anzulegen, die Web-App für das Testen unmittelbar auszuführen und den Code mit Git und GitHub für die Versionsverwaltung zu integrieren. Eine interaktive Konsole und eine integrierte Hilfe erleichtern den Arbeitsprozess.

5.3 Versionskontrolle

Versionskontrolle ist aus vielerlei Gründen für den Softwareentwicklungsprozess unerlässlich. Sie ermöglicht es,

- Versionen zu verwalten,
- Bugs durch Differenzbildung zwischen verschiedenen Versionen zu finden und zu beheben sowie
- den Projektfortschritt zeitlich und inhaltlich nachzuvollziehen.

Für die Versionskontrolle wurde Git in Verwendung mit GitHub eingesetzt. In Git werden inkrementelle Änderungen durch sogenannte Commits erfasst. Jeder Commit ist dabei mit einem Kommentar versehen. Durch sogenannte Branches können verschiedene Personen gleichzeitig zum Projekt beitragen oder verschiedene Features gleichzeitig entwickeln. Branches können wieder zusammengeführt werden (Merging). Ein Ordner, der mit Git initialisiert wurde, wird als Repository bezeichnet. Repositories können im Internet verfügbar gemacht werden und dann auf beliebigen Computern heruntergeladen werden.

Die Online-Plattform GitHub erleichtert die Kollaboration über Git. Sie stellt einen Ablageort für das Repository bereit und unterstützt den Arbeitsprozess durch ein Ticket-System (Issues). In diesem können Fehler und Verbesserungsvorschläge gemeldet werden. Die Issues können darüber hinaus als Notizblock für geplante Features verwendet werden.

6 Umsetzung

Die folgenden Abschnitte stellen die Umsetzung der Web-Applikation sowohl aus der anwendungsorientierten als auch der technikorientierten Sichtweise dar.

6.1 Anwendungsorientierte Sichtweise

Die Web-Applikation ist ein Dashboard. Dieses besitzt eine Sidebar, eine Navbar und einen Body.

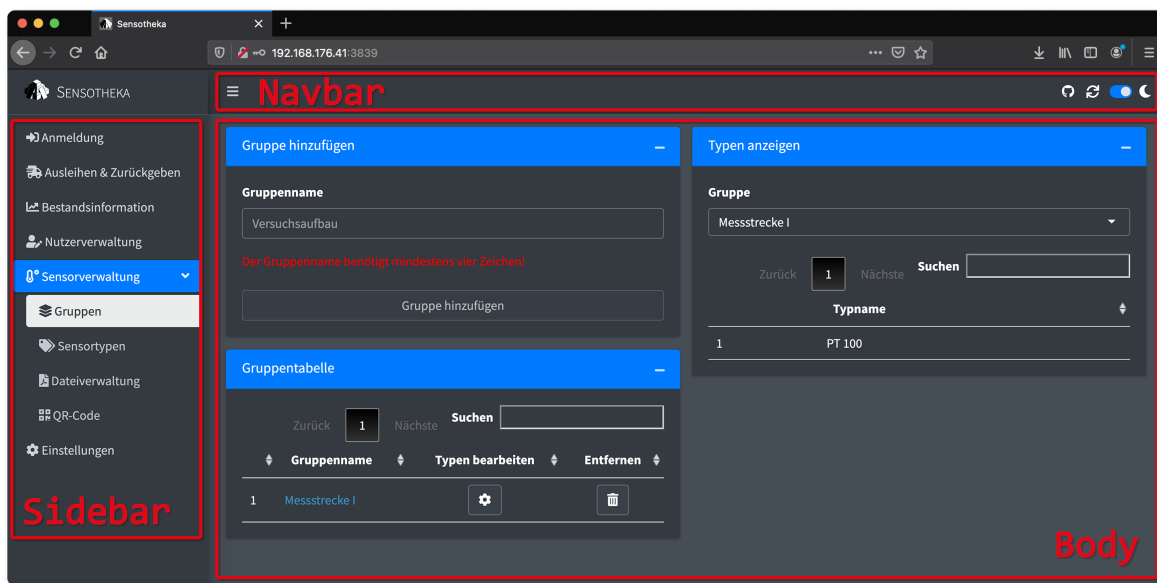


Abbildung 1: Übersicht über das Dashboard

6.1.1 Sidebar

In der Sidebar kann über einen Reiter die Funktionalität ausgewählt werden, die im Body dargestellt wird. Die Anzahl an Auswahlmöglichkeiten hängt vom Anmeldestatus und der Benutzerrolle ab.

6.1.2 Navbar

Die Navbar enthält:

- einen Link zum GitHub-Repository, das den Quellcode der Web-Applikation enthält,
- einen Reload-Button, mit dem eine Aktualisierung der Daten erzwungen werden kann,
- einen Toggle, mit dem zwischen Tag- und Nachtmodus gewechselt werden kann.

6.1.3 Anmeldung

Dieser Reiter verändert sich in Abhängigkeit des Anmeldestatus.

6.1.3.1 Status: Nicht angemeldet Registrierte Benutzer können sich durch Angabe ihres Benutzernamens und ihres Passwortes anmelden. Nicht-registrierte Benutzer müssen sich von einem Moderator oder Administrator (Wissenschaftliche Mitarbeiter) registrieren lassen.

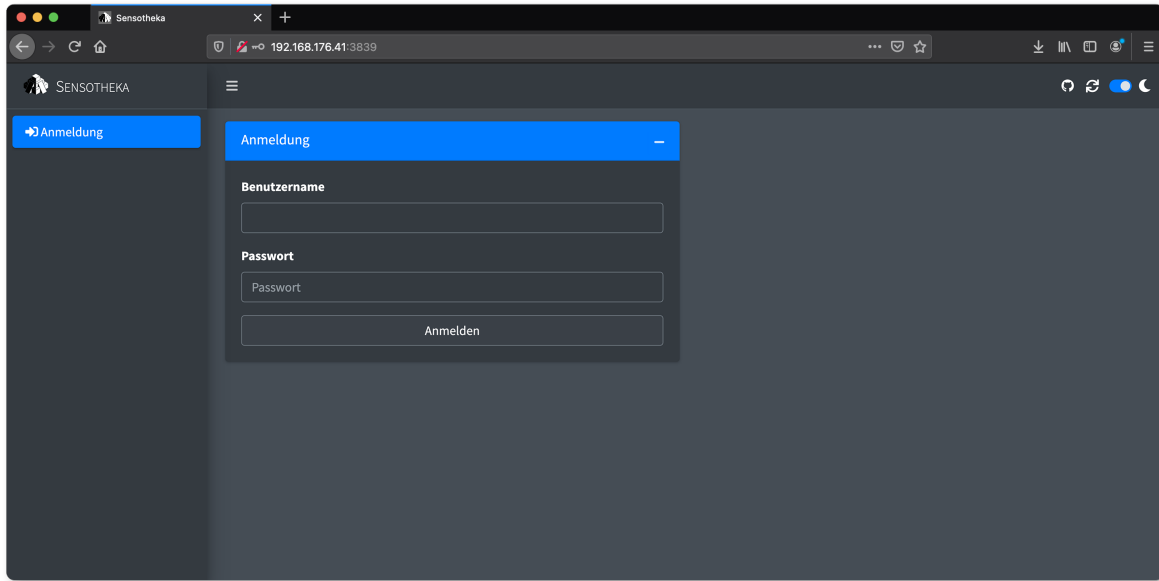


Abbildung 2: Anmeldung - nicht angemeldet

6.1.3.2 Status: Angemeldet Nach erfolgreicher Anmeldung wird der Benutzer mit Informationen über sein Nutzungsverhalten versorgt. Dargestellt werden:

- der Benutzername und der Benutzerstatus,
- die Dauer seit der momentanen Anmeldung,
- die Dauer seit der letzten Anmeldung,
- die Anzahl der Anmeldungen.

Benutzer können sich zudem abmelden.

6.1.4 Ausleihen & Zurückgeben

Die Ausleihe und Rückgabe setzt sich aus drei Abschnitten zusammen. Im ersten Abschnitt *Ausleihen & Zurückgeben* kann die konkrete Operation vorgenommen werden. Zusätzliche Informationen stellen die Abschnitte *Gruppen des ausgewählten Typs* und *Dateien* bereit.

6.1.4.1 Ausleihen & Zurückgeben Zunächst muss ein Typ ausgewählt werden. Wenn die Web-Applikation über einen QR-Code aufgerufen wurde, ist der zum QR-Code gehörende Typ bereits ausgewählt. Im nächsten Schritt muss ein zugehöriger Untertyp ausgewählt werden. Die konkrete Operation kann durch Klicken auf einen der verfügbaren Buttons angestoßen werden. Im sich darauf öffnenden Dialog muss die Menge angegeben und die Ausführung der Operation bestätigt werden.

Administratoren können zusätzlich den ausführenden Nutzer auswählen, um Operationen für diesen durchzuführen. Das ist zum Beispiel notwendig, wenn ein Student das Fachgebiet verlassen hat und es versäumt hat, alle ausgeliehenen Sensoren zurückzugeben.

Es können bis zur maximal verfügbaren Menge Elemente ausgeliehen werden.

Es können bis zur maximal ausgeliehenen Menge Elemente zurückgegeben werden.

Es können bis zur maximal im Lager verfügbaren Menge Elemente abgeschrieben werden. Dies ist zum Beispiel notwendig, wenn ein Element ausfällt oder verloren geht.

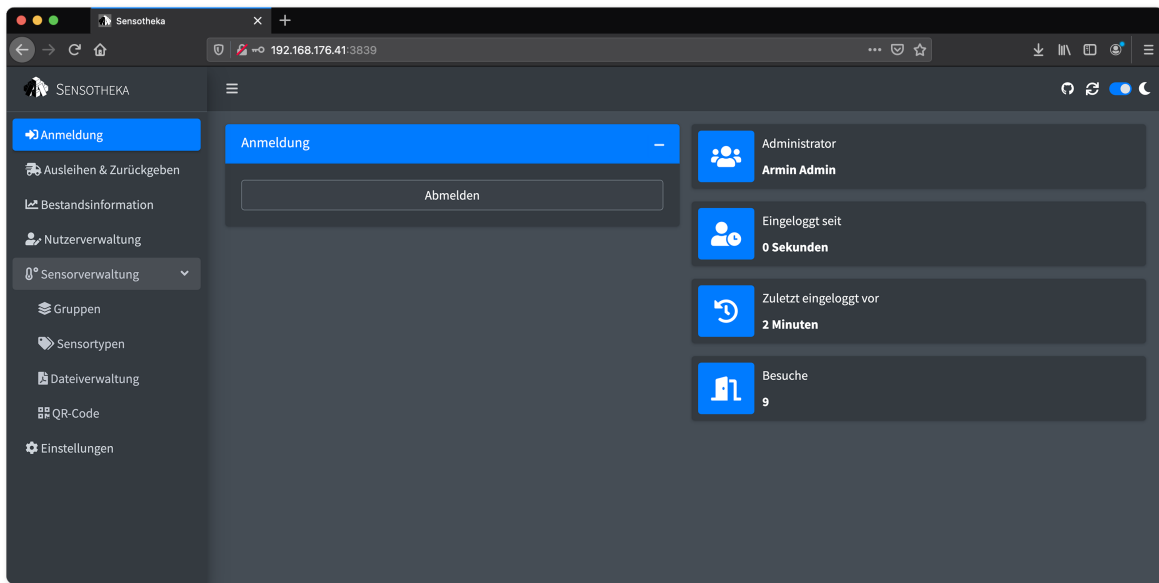


Abbildung 3: Anmeldung - angemeldet

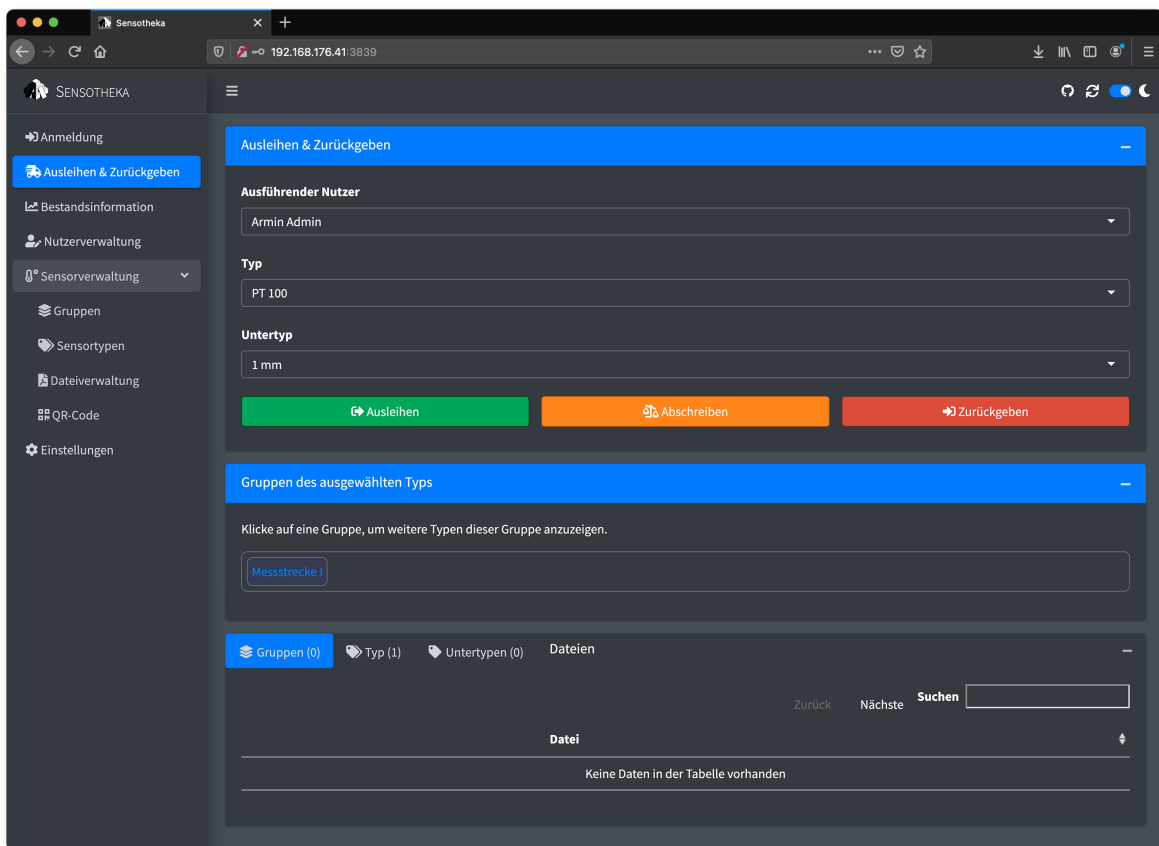


Abbildung 4: Übersicht: Ausleihen & Zurückgeben

The image shows a software interface titled "Ausleihen & Zurückgeben". It contains three dropdown menus for selection: "Ausführender Nutzer" with "Armin Admin" selected, "Typ" with "PT 100" selected, and "Untertyp" with "1 mm" selected. Below these are three action buttons: a green "Ausleihen" button, an orange "Abschreiben" button, and a red "Zurückgeben" button.

Abbildung 5: Abschnitt: Ausleihen & Zurückgeben

6.1.4.2 Gruppen des ausgewählten Typs Alle Gruppen des ausgewählten Typs werden aufgelistet. Durch Klicken auf eine Gruppe öffnet sich ein Dialog, in dem alle Typen dieser Gruppe angezeigt werden. Durch Klicken auf einen Typen wird dieser Typ für eine weitere Operation ausgewählt. Somit können alle Elemente einer Gruppe (zum Beispiel ein Versuchsaufbau) komfortabel hintereinander ausgeliehen / zurückgegeben werden.

6.1.4.3 Dateien Hier werden alle dem ausgewählten Typ zugeordneten Dateien angezeigt. Dabei ist die Anzeige in drei Reiter gegliedert:

- *Gruppen*: Alle Dateien, die Informationen zu Gruppen des ausgewählten Typs enthalten
- *Typ*: Alle Dateien, die Informationen zum ausgewählten Typ enthalten
- *Untertypen*: Alle Dateien, die Informationen zu Untertypen des ausgewählten Typs enthalten. Es ist zu beachten, dass die Wahl des Untertypen keinen Einfluss auf die angezeigten Dateien hat, da Dateien für alle Untertypen dargestellt werden.

Die Ziffer neben dem Reitertitel gibt Auskunft darüber, wie viele Dateien in der jeweiligen Kategorie vorhanden sind. Indem auf einen Dateinamen geklickt wird, öffnet sich das zugehörige PDF in einem separaten Browserfenster oder wird über den PDF-Viewer angezeigt. Alle Dateien einer Kategorie können in einem Archiv (.zip) heruntergeladen werden.

6.1.5 Bestandsinformation

Die Bestandsinformation gliedert sich in zwei Abschnitte. In der Bestandsübersicht werden Informationen zum Lagerbestand von Sensoren dargestellt, wohingegen die Ausleihübersicht Informationen über ausgeliehene Sensoren enthält.

6.1.5.1 Bestandsübersicht Zunächst muss ein Typ ausgewählt werden. Für diesen Typen werden tabellarisch alle Untertypen dieses Typs aufgelistet. Die Spalte *Verfügbar* enthält die gelagerte Menge, wohingegen die Spalte *Maximal verfügbar* die maximal gelagerte Menge (kein Sensor dieses Untertypen ausgeliehen) enthält. Über die Checkbox *Nur kritische Bestände anzeigen* kann ausgewählt werden, ob nur Untertypen angezeigt werden sollen, für die die verfügbare Menge kleiner als ihr kritischer Bestand ist. Der kritische Bestand eines Untertypen kann von einem im Reiter *Sensortypen* festgelegt werden.

6.1.5.2 Ausleihübersicht Die Ausleihübersicht besteht aus vier Reitern, die unterschiedliche Fragestellen in Bezug auf ausgeliehene Sensoren beantworten.

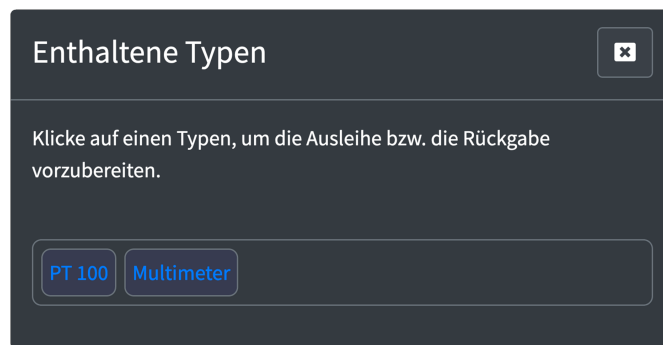
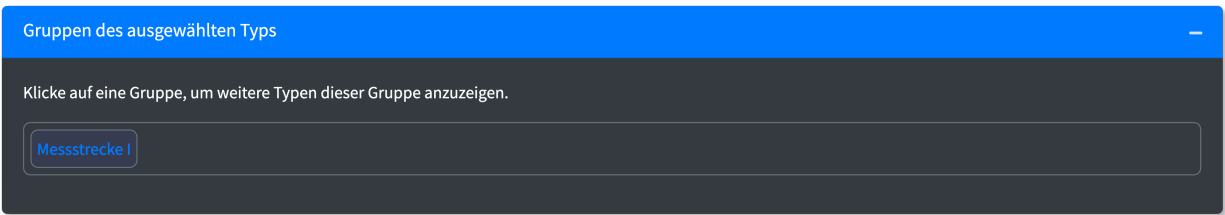


Abbildung 6: Abschnitt: Gruppen des ausgewählten Typs

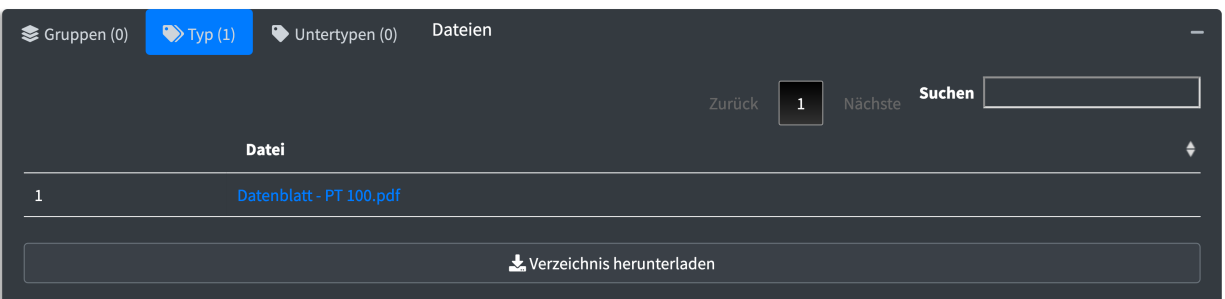


Abbildung 7: Abschnitt: Dateien

Reiter	Beschreibung
Gesamt	Übersicht über alle Untertypen mit einer gegenwärtig ausgeliehenen Menge größer Null.
Nach Untertyp	Übersicht über alle Nutzer, die den gewählten Untertypen zurzeit ausgeliehen haben. Hiermit kann herausgefunden werden, wer Ansprechpartner ist, falls alle Elemente eines Untertyps ausgeliehen sind.
Nach Nutzer	Übersicht über alle Untertypen, die der gewählte Nutzer zurzeit ausgeliehen hat.
Transaktionen	Übersicht über alle Transaktionen, die jemals im Rahmen der Sensorverwaltung stattgefunden haben. Ein Nutzer sieht ausschließlich seine eigenen Transaktionen. Ein sieht alle Transaktionen. Dazu gehören neben Ausleih- und Rückgabeoperationen auch Abschreibungen und sonstige Mengenänderungen.

6.1.6 Nutzerverwaltung

In der *Nutzerverwaltung* können neue Nutzer hinzugefügt und in der *Nutzertabelle* verwaltet werden.

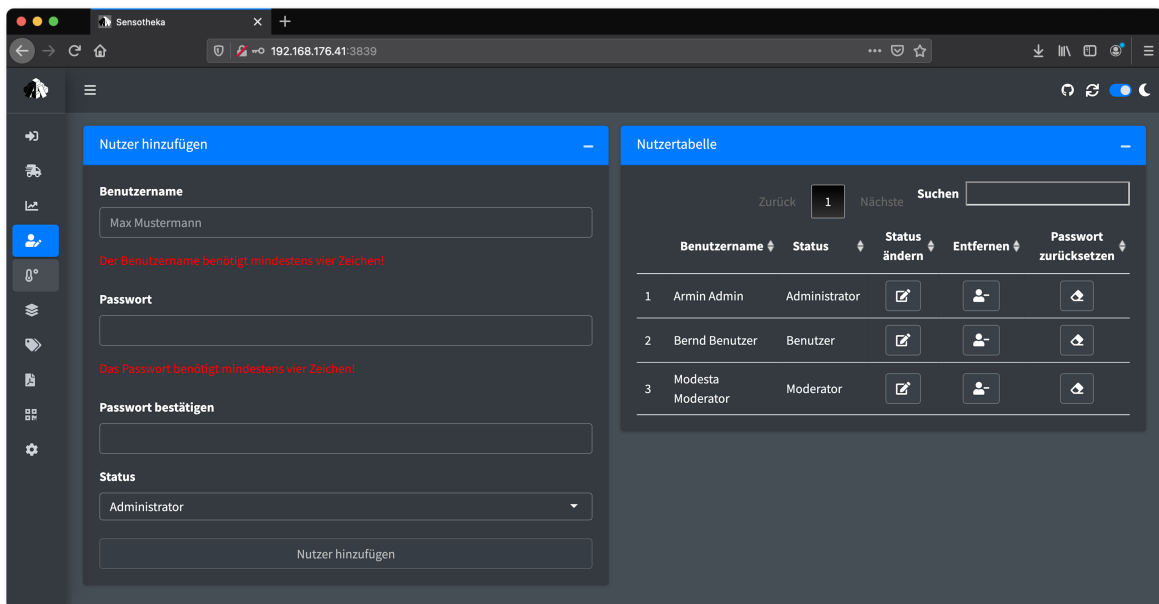


Abbildung 9: Übersicht: Nutzerverwaltung

6.1.6.1 Nutzer hinzufügen Neue Nutzer können unter Angabe eines Benutzernamens und eines zur Sicherheit zweimal anzugebenden Passwortes hinzugefügt werden. Ein kann ausschließlich hinzufügen, wohingegen ein Benutzer mit jeder Rolle hinzufügen kann.

6.1.6.2 Nutzertabelle In der Nutzertabelle können Nutzer verwaltet werden. Neben dem Benutzernamen und dem gegenwärtigen Status gibt es drei Spalten, die Buttons enthalten, mit denen Eigenschaften eines Nutzer angepasst werden können.

Spalte	Beschreibung
Status ändern	Dieser Button öffnet einen Dialog, in dem ein neuer Status für den Nutzer ausgewählt werden kann.
Entfernen	Dieser Button öffnet einen Dialog, in dem bestätigt werden muss, dass der ausgewählte Nutzer gelöscht werden soll. Es ist zu beachten, dass diese Operation nicht rückgängig gemacht werden kann. Transaktionen, die der ausgewählte Nutzer vorgenommen hat, bleiben jedoch weiterhin erhalten. Es können nur Nutzer gelöscht werden, die zurzeit keine Sensoren ausgeliehen haben. Falls der Nutzer nicht in der Lage ist, die Sensoren eigenständig zurückzugeben, kann der die ausgeliehenen Sensoren im Reiter <i>Ausleihen & Zurückgeben</i> für diesen Nutzer zurückgeben. Ein kann nur entfernen, die er selbst hinzugefügt hat.
Passwort zurücksetzen	Dieser Button öffnet einen Dialog, in dem bestätigt werden muss, dass das Passwort des Nutzers zurückgesetzt werden soll. Das Passwort wird auf das Standardpasswort <i>1234</i> zurückgesetzt und sollte vom Nutzer sofort danach im Reiter <i>Einstellungen</i> geändert werden.

6.1.7 Gruppen

Der Reiter *Gruppen* erlaubt es, neue Gruppen hinzuzufügen und bestehende Gruppen zu verwalten.

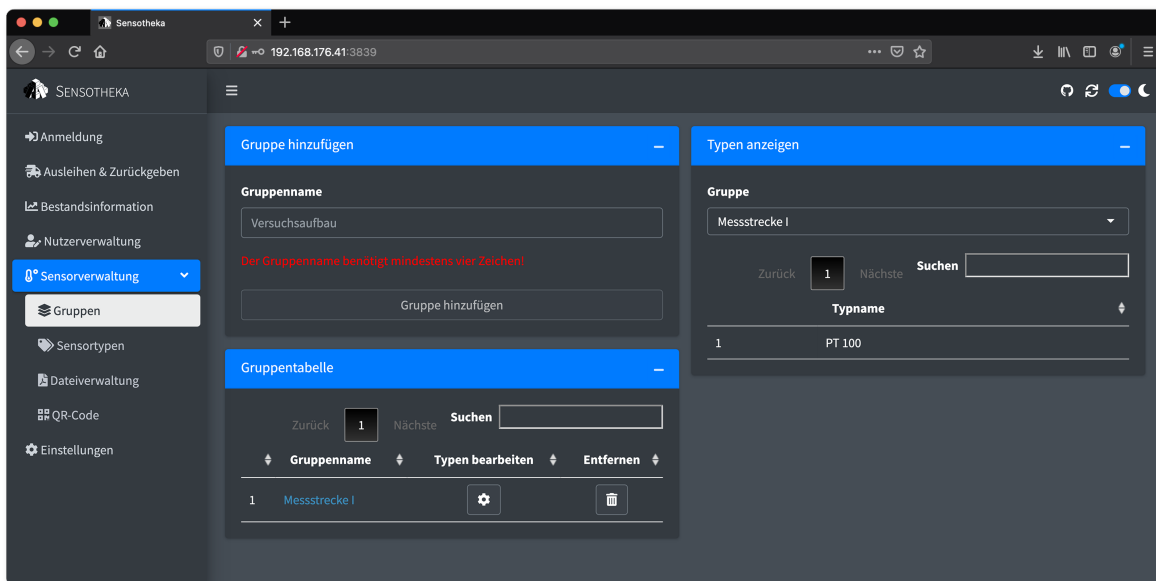


Abbildung 10: Übersicht: Nutzerverwaltung

6.1.7.1 Gruppe hinzufügen Eine neue Gruppe kann durch Angabe eines Gruppennamens hinzugefügt werden.

6.1.7.2 Gruppentabelle In der *Gruppentabelle* können Gruppen bearbeitet werden.

Spalte	Beschreibung
Gruppenname	Ein Klick auf einen Gruppennamen öffnet einen Dialog, in dem der Gruppenname angepasst werden kann.
Typen bearbeiten	Dieser Button öffnet einen Dialog, in dem die zur ausgewählten Gruppe zugehörigen Typen bearbeitet werden können.
Entfernen	Dieser Button öffnet einen Dialog, in dem bestätigt werden muss, dass die ausgewählte Gruppe gelöscht werden soll. Es ist zu beachten, dass diese Operation nicht rückgängig gemacht werden kann.

6.1.7.3 Typen anzeigen Zunächst muss eine Gruppe ausgewählt werden. In der Tabelle werden alle zur ausgewählten Gruppe zugehörigen Typen dargestellt.

6.1.8 Sensortypen

Der Reiter *Sensortypen* erlaubt es, neue Typen und Untertypen hinzuzufügen und bestehende Typen und Untertypen zu verwalten.

6.1.8.1 Typ hinzufügen Ein neuer Typ kann durch Angabe eines Typnamens hinzugefügt werden.

6.1.8.2 Typtabelle In der *Typtabelle* können Typen bearbeitet werden.

Spalte	Beschreibung
Typname	Ein Klick auf einen Typnamen öffnet einen Dialog, in dem der Typname angepasst werden kann.
Gruppen bearbeiten	Dieser Button öffnet einen Dialog, in dem die zum ausgewählten Typ zugehörigen Gruppen bearbeitet werden können.
Entfernen	Dieser Button öffnet einen Dialog, in dem bestätigt werden muss, dass der ausgewählte Typ gelöscht werden soll. Es ist zu beachten, dass diese Operation nicht rückgängig gemacht werden kann. Ein Typ kann nur gelöscht werden, wenn kein Element seiner Untertypen ausgeliehen ist. Transaktionen, die den ausgewählten Typen betreffen, bleiben weiterhin erhalten. Das Entfernen eines Typen schließt das Entfernen aller Untertypen dieses Typen ein.

6.1.8.3 Gruppen anzeigen Zunächst muss ein Typ ausgewählt werden. In der Tabelle werden alle zum ausgewählten Typen zugehörigen Gruppen dargestellt.

6.1.8.4 Untertypen bearbeiten Zunächst muss ein Typ ausgewählt werden. In der Tabelle können Untertypen des ausgewählten Typen bearbeitet werden.

Spalte	Beschreibung
Untertypname	Ein Klick auf einen Untertypnamen öffnet einen Dialog, in dem der Untertypname angepasst werden kann.
Menge	Ein Klick auf die maximal verfügbare Menge öffnet einen Dialog, in dem die maximal verfügbare Menge des Untertypen angepasst werden kann. Es ist zu beachten, dass die maximal verfügbare Menge nicht auf einen Wert gesetzt werden kann, der kleiner der Anzahl gegenwärtig ausgeliehener Elemente ist.
Kritischer Bestand	Ein Klick auf den kritischen Bestand öffnet einen Dialog, in dem der kritische Bestand des Untertypen angepasst werden kann. Der kritische Bestand kann als Filterkriterium in der <i>Ausleihübersicht</i> des Reiters <i>Bestandsinformation</i> verwendet werden.
Entfernen	Dieser Button öffnet einen Dialog, in dem bestätigt werden muss, dass der ausgewählte Untertyp gelöscht werden soll. Es ist zu beachten, dass diese Operation nicht rückgängig gemacht werden kann. Ein Untertyp kann nur gelöscht werden, wenn keine Elemente von diesem ausgeliehen sind. Transaktionen, die den ausgewählten Untertypen betreffen, bleiben weiterhin erhalten.

6.1.9 Dateiverwaltung

In der *Dateiverwaltung* können PDF-Dateien als Informationsmaterial in den Reitern *Gruppen*, *Typen* und *Untertypen* hochgeladen werden. Die Reiter unterscheiden sich ausschließlich in der Auswahl des zu beschreibenden Objektes. Durch Klick auf *Datei hochladen* kann eine PDF-Datei auf dem lokalen Dateisystem ausgewählt werden. Hochgeladene Dateien werden in einer Tabelle angezeigt.

Spalte	Beschreibung
Datei	Ein Klick auf einen Dateinamen öffnet einen Dialog, in dem der Dateiname angepasst werden kann.
Herunterladen	Dieser Button lädt die Datei herunter.
Löschen	Dieser Button öffnet einen Dialog, in dem bestätigt werden muss, dass die ausgewählte Datei gelöscht werden soll. Es ist zu beachten, dass diese Operation nicht rückgängig gemacht werden kann.

6.1.10 QR-Code

6.1.11 Einstellungen

6.1.11.1 Benutzernamen ändern Um den eigenen Benutzernamen zu ändern, muss ein neuer Benutzername eingegeben werden. Nach Eingabe des Passworts und Bestätigen wird der Benutzername geändert.

6.1.11.2 Passwort ändern Um das eigene Passwort zu ändern, muss ein neues Passwort zweimal eingegeben und bestätigt werden.

6.2 Technikorientierte Sichtweise

7 Diskussion

8 Ausblick