



MEDIAVERSE

Software-Entwicklungspraktikum (SEP)

Sommersemester 2023

Technischer Entwurf

Auftraggeber

Technische Universität Braunschweig

Institut für Robotik und Prozessinformatik - IRP

Mühlenpfordtstraße 23

38106 Braunschweig

Betreuer: Dr. Bertold Bongardt, Heiko Donat, Sven Tittel, Christopher Lösch

Auftragnehmer:

Name	E-Mail-Adresse
Michèle Eger	m.eger@tu-braunschweig.de
Fahd Ferjani	f.ferjani@tu-braunschweig.de
Yassine Kechiche	y.kechiche@tu-braunschweig.de
Yiğit Kemal Çağlar	y.caglar@tu-braunschweig.de
Oussema Ben Smida	o.ben-smida@tu-braunschweig.de
Subing Shen	subing.shen@tu-braunschweig.de
Quynh Tran	quynh.tran@tu-braunschweig.de
Theodore Zebua	t.zebua@tu-braunschweig.de

Braunschweig, 28. Juni 2023

Bearbeiterübersicht

Kapitel	Autoren	Kommentare
1	Yassine, Fahd	...
1.1	Yassine, Fahd	...
2	Oussema, Michèle, Yigit, Theodore	...
2.1	Oussema	...
2.2	Oussema	...
2.3	Michèle	...
2.4	Michèle	...
2.5	Michèle	...
2.6	Yigit	...
2.7	Yigit	...
2.8	Oussema	...
2.9	Yigit	...
2.10	Theodore	...
2.11	Theodore	...
2.12	Theodore	...
3	Yassine, Fahd	...
3.1	Yassine, Fahd	...
3.2	Yassine	...
3.3	Oussema, Theodore	...
4	Fahd	...
5	Quynh, Michèle	...
5.1	Michèle, Quynh	...
5.2	Michèle, Quynh	...
5.3	Yigit	...
5.4	Subing	...
5.5	Subing	...
5.6	Theodore	...
5.7	Theodore	...
6	Quynh, Subing	...
6.1	Quynh, Subing	...
6.2	Subing, Quynh	...
7	Yassine	...
8	Subing, Quynh, Yigit	...

9	Quynh, Michèle	...
9.1	Quynh, Michèle	...
9.2	Quynh, Michèle	...
9.3	Quynh, Michèle	...
10	Fahd, Yassine	...

Inhaltsverzeichnis

1	Einleitung	9
1.1	Projektübersicht	9
1.1.1	Benutzer	10
1.1.2	Mediathekar	12
1.1.3	Admin	14
1.2	Projektdetails	16
1.2.1	Verwaltung von Favoritenliste und teilen von Ressourcen	16
2	Analyse der Produktfunktionen	18
2.1	Analyse von Funktionalität F10 : Registrierung	19
2.2	Analyse von Funktionalität F20 : Login	20
2.3	Analyse von Funktionalität F30 : Suchfunktion	21
2.4	Analyse von Funktionalität F40 : Verfügbarkeit, Ausleihen und Ausleih-Verlängerung	22
2.5	Analyse von Funktionalität F50 : Herunterladen	24
2.6	Analyse von Funktionalität F60 : Hinzufügen, Bearbeiten, Löschen von Ressourcen	25
2.7	Analyse von Funktionalität F70 : Favoritenlisten	26
2.8	Analyse von Funktionalität F80 : Spracheinstellung	27
2.9	Analyse von Funktionalität F90 : Verlauf der (eigenen) Aktionen	28
2.10	Analyse von Funktionalität F100 : Anlegen von Benutzern und Mediathekaren .	29
2.11	Analyse von Funktionalität F110 : Editieren und Löschen von Benutzern und Mediathekaren	30
2.12	Analyse von Funktionalität F120 : QR-Code	31
3	Resultierende Softwarearchitektur	32
3.1	Komponentenspezifikation	32
3.2	Schnittstellenspezifikation	36
3.3	Protokolle für die Benutzung der Komponenten	40
3.3.1	Protokoll für $\langle C10 \rangle$: Benutzerverwaltung	41
3.3.2	Protokoll für $\langle C20 \rangle$: Ressourcenverwaltung	42
3.3.3	Protokoll für $\langle C30 \rangle$: Suchfunktion	43
3.3.4	Protokoll für $\langle C40 \rangle$: Ausleihverwaltung	44
3.3.5	Protokoll für $\langle C50 \rangle$: Favoritenverwaltung	45
3.3.6	Protokoll für $\langle C60 \rangle$: Benutzeraktivitätenverfolgung	45

3.3.7	Protokoll für $\langle C70 \rangle$: Sprachverwaltung	46
4	Verteilungsentwurf	47
5	Implementierungsentwurf	50
5.1	Implementierung von Komponente $\langle C10 \rangle$: Benutzerverwaltung	50
5.1.1	Paket-/Klassendiagramm	50
5.1.2	Erläuterung	51
5.2	Implementierung von Komponente $\langle C20 \rangle$: Ressourcenverwaltung	53
5.2.1	Paket-/Klassendiagramm	53
5.2.2	Erläuterung	53
5.3	Implementierung von Komponente $\langle C30 \rangle$: Suchfunktion	57
5.3.1	Paket-/Klassendiagramm	57
5.3.2	Erläuterung	58
5.4	Implementierung von Komponente $\langle C40 \rangle$: Ausleihverwaltung	59
5.4.1	Paket-/Klassendiagramm	59
5.4.2	Erläuterung	59
5.5	Implementierung von Komponente $\langle C50 \rangle$: Favoritenverwaltung	61
5.5.1	Paket-/Klassendiagramm	61
5.5.2	Erläuterung	62
5.6	Implementierung von Komponente $\langle C60 \rangle$: Benutzeraktivitätenverfolgung	63
5.6.1	Paket-/Klassendiagramm	63
5.6.2	Erläuterung	63
5.7	Implementierung von Komponente $\langle C70 \rangle$: Sprachverwaltung	64
5.7.1	Paket-/Klassendiagramm	65
5.7.2	Erläuterung	65
6	Datenmodell	67
6.1	Diagramm	67
6.2	Erläuterung	70
6.2.1	Verwaltung der Konten	73
7	Konfiguration	74
8	Änderungen gegenüber Fachentwurf	76
9	Erfüllung der Kriterien	77
9.1	Musskriterien	77
9.2	Sollkriterien	80
9.3	Kannkriterien	80

Abbildungsverzeichnis

1.1	Statechart: Benutzer	11
1.2	Statechart: Mediathekar	13
1.3	Statechart: Admin	15
1.4	Aktivitätsdiagramm(UML): Favoritenliste	17
2.1	Sequenzdiagramm Registrierung	19
2.2	Sequenzdiagramm Login	20
2.3	Sequenzdiagramm Suchfunktion	21
2.4	Sequenzdiagramm Verfügbarkeit und Ausleihen	22
2.5	Sequenzdiagramm Ausleih-Verlängerung	23
2.6	Sequenzdiagramm Download von Ressourcen	24
2.7	Sequenzdiagramm Hinzufügen, Bearbeiten, Löschen von Ressourcen	25
2.8	Sequenzdiagramm Favoritenlisten	26
2.9	Sequenzdiagramm Spracheinstellung	27
2.10	Sequenzdiagramm Verlauf der (eigenen) Aktionen	28
2.11	Sequenzdiagramm Anlegen von Benutzern und Mediathekaren	29
2.12	Sequenzdiagramm Editieren und Löschen von Benutzern und Mediathekaren	30
2.13	Sequenzdiagramm QR-Code	31
3.1	Komponentendiagramm	33
3.2	Zustandsdiagramm der Komponente $\langle C10 \rangle$	41
3.3	Zustandsdiagramm der Komponente $\langle C20 \rangle$	42
3.4	Zustandsdiagramm der Komponente $\langle C30 \rangle$	43
3.5	Zustandsdiagramm der Komponente $\langle C40 \rangle$	44
3.6	Zustandsdiagramm der Komponente $\langle C50 \rangle$	45
3.7	Zustandsdiagramm der Komponente $\langle C60 \rangle$	45
3.8	Zustandsdiagramm der Komponente $\langle C70 \rangle$	46
4.1	Verteilungsdiagramm	49
5.1	Klassendiagramm für Komponente $\langle C10 \rangle$	50
5.2	Klassendiagramm für Komponente $\langle C20 \rangle$	53
5.3	Klassendiagramm für Komponente $\langle C30 \rangle$	57
5.4	Klassendiagramm für Komponente $\langle C40 \rangle$	59

5.5	Klassendiagramm für Komponente $\langle C50 \rangle$	61
5.6	Klassendiagramm für Komponente $\langle C60 \rangle$	63
5.7	Klassendiagramm für Komponente $\langle C70 \rangle$	65
6.1	Klassendiagramm: Projektübersicht	69

1 Einleitung

Dieses Dokument stellt den Technischen Entwurf für das Mediaverse-Projekt dar. Es baut auf dem vorherigen Fachentwurf auf und bietet eine detaillierte Erläuterung der Funktionalitäten aller Komponenten. Dabei werden das genaue Verhalten der Funktionen und der Aufbau der einzelnen Komponenten mithilfe zahlreicher Diagramme veranschaulicht. Der Technische Entwurf präsentiert die im Projekt verwendete Software-Architektur, um dem Auftraggeber einen detaillierten Einblick in das geplante System zu geben.

1.1 Projektübersicht

Das Mediaverse-Projekt wird im Rahmen des Softwareentwicklungspraktikums durchgeführt, um eine umfassende Plattform zur Verwaltung und Bereitstellung von Medieninhalten zu entwickeln. Die Plattform ermöglicht es Benutzern, verschiedene Medienressourcen wie Bilder, Videos, Audiodateien, Textdokumente und sogar Geräte zu suchen, auszuwählen und zu nutzen.

Mediaverse bietet eine breite Palette von Medieninhalten und Geräten, die den Benutzern zur Verfügung stehen. Sie können nach bestimmten Medien suchen, Inhalte streamen oder herunterladen, persönliche Favoritenlisten erstellen und auch auf Geräte zugreifen. Die Benutzeroberfläche wurde benutzerfreundlich und ansprechend gestaltet, um eine intuitive Navigation durch den Medienkatalog und die Geräteliste zu ermöglichen.

Für die Entwicklung der Mediaverse-Anwendung werden moderne Webtechnologien und Frameworks wie Svelte, Django und PostgreSQL eingesetzt. RESTful APIs werden verwendet, um die Kommunikation zwischen Frontend und Backend zu ermöglichen. Das Ziel ist es, den Benutzern eine benutzerfreundliche Erfahrung zu bieten und ihre Anforderungen an eine Medienplattform zu erfüllen.

Mediaverse bietet drei Arten von Benutzern: Benutzer, Mediathekar und Administrator. Jeder Benutzertyp hat spezifische Funktionen und Berechtigungen innerhalb der Plattform. Die Statecharts in den kommenden Abschnitten zeigen die Hauptfunktionen der verschiedenen Benutzertypen in Mediaverse.

1.1.1 Benutzer

Der Statechart aus Abbildung 1.1 veranschaulicht die Funktionsweise von Mediaverse für normale Benutzer. Es zeigt die verschiedenen Zustände, die ein Benutzer während der Nutzung der Plattform durchläuft.

Nach dem Öffnen der Mediaverse-Website gelangt der Benutzer zur "Mediaverse Einloggen geöffnet". Hier kann er entweder "Einloggen" nutzen, um sich anzumelden, oder "Konto erstellen" besuchen, um ein neues Konto zu erstellen.

Es ist jedoch zu beachten, dass nach der Registrierung der Benutzer auf die Freigabe durch den Administrator warten muss, bevor er sich einloggen kann. Sobald das Konto durch den Administrator aktiviert wurde, kann der Benutzer anschließend auf das System zugreifen.

Nach erfolgreichem Einloggen befindet sich er im Zustand "In Mediaverse eingeloggt". Dieser Zustand umfasst verschiedene Unterzustände, die die Funktionen und Seiten von Mediaverse repräsentieren.

Der erste Unterzustand ist "Start geöffnet". Hier kann der Benutzer eine Suche nach Medienressourcen durchführen. Nach der Suche werden die Ergebnisse. Der Benutzer kann die Ergebnisse filtern, eine neue Suche durchführen und interessante Elemente zur Favoritenliste hinzufügen oder ausgewählte Ressourcen näher betrachten und nutzen, sofern verfügbar.

Ein weiterer Unterzustand ist "Favoriten Liste geöffnet". Hier werden dem Benutzer die Elemente seiner Favoritenliste angezeigt. Er kann einzelne Elemente bearbeiten, löschen oder teilen. Der Zustand "Ressourcen Liste geöffnet" ermöglicht es dem Benutzer, verfügbare Ressourcen anzuzeigen. Er kann zwischen der Anzeige der aktuellen verwendeten Ressourcen und dem Verlauf der bereits genutzten Ressourcen wählen. Der Zustand "Profil geöffnet" ermöglicht dem Benutzer die Anzeige und Bearbeitung seiner Konto Einstellungen.

Es ist wichtig zu beachten, dass der eingeloggte Benutzer jederzeit zwischen den verschiedenen Seiten und Zuständen wechseln kann, indem er die entsprechenden Optionen im Menü auswählt. Diese Funktionalitäten und Zustände bilden die Grundlage für die Nutzung von Mediaverse.

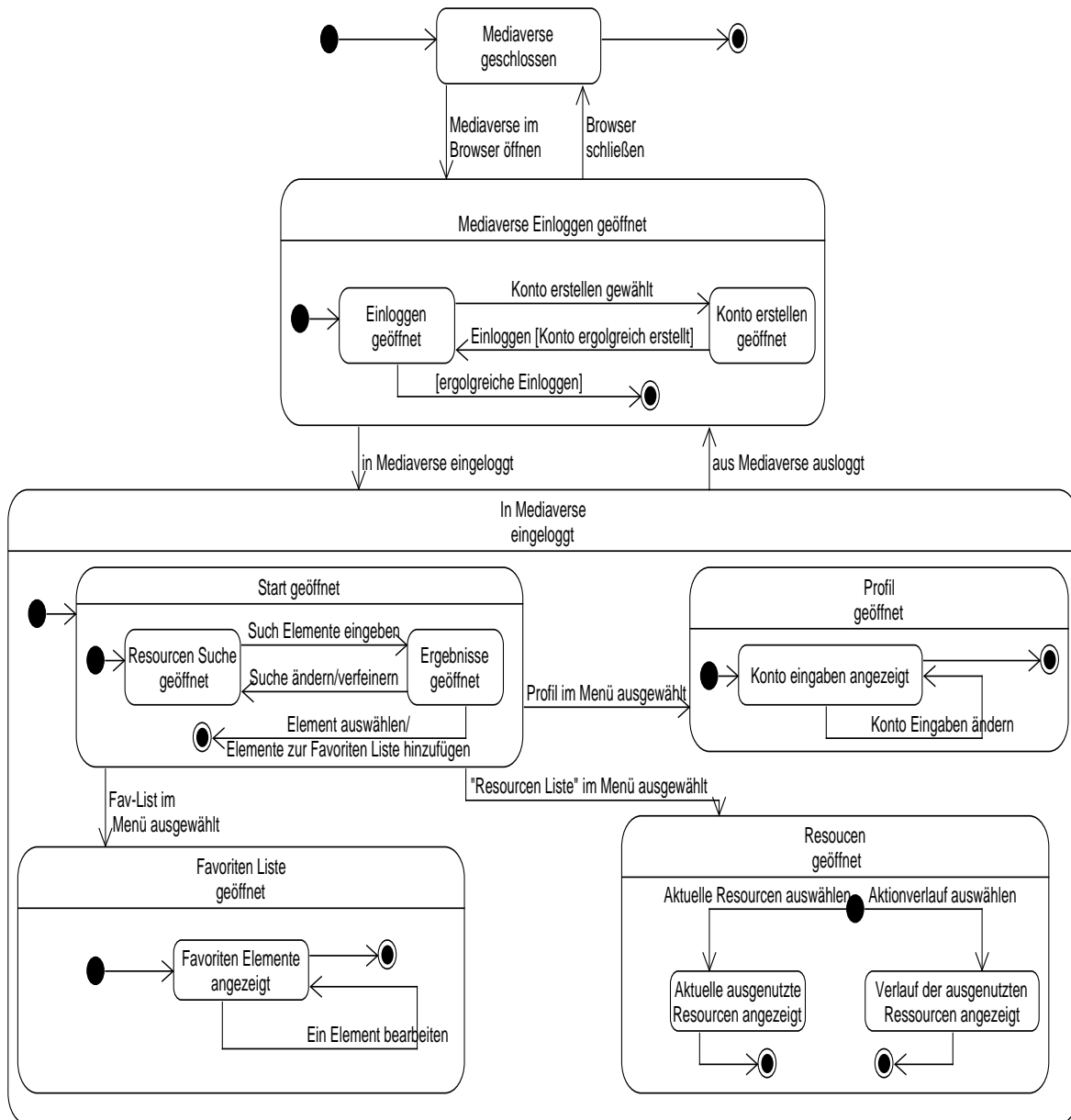


Abbildung 1.1: Statechart: Benutzer

1.1.2 Mediathekar

Der Statechart in Abbildung 1.2 veranschaulicht die Funktionsweise von Mediaverse für Mediathekare. Es zeigt die verschiedenen Zustände, die ein Mediathekar während der Nutzung der Plattform durchläuft. Dieses Statechart ergänzt den allgemeinen Statechart für normale Benutzer, da der Mediathekar auch ein normaler Benutzer ist und somit über alle Funktionen eines normalen Benutzers verfügt. Der zusätzliche Statechart spezifiziert die Zustände und Übergänge, die speziell für die Ressourcenverwaltung des Mediathekars relevant sind.

Der Statechart beginnt mit dem Zustand "In Mediaverse als Mediathekar eingeloggt". Dieser Zustand repräsentiert den eingeloggten Zustand des Mediathekars in der Plattform. Um in den Zustand "Ressourcen Verwaltung geöffnet" zu gelangen, wird die Bedingung "Ressourcenverwaltung" im Menü ausgewählt. Dies öffnet die Seite für die Ressourcenverwaltung und ermöglicht dem Mediathekar, verschiedene Funktionen im Zusammenhang mit der Verwaltung von Ressourcen durchzuführen.

Der Zustand "Ressourcen Verwaltung geöffnet" enthält weitere spezifische Zustände, die mit der Ressourcenverwaltung zusammenhängen.

Der Ausgangszustand innerhalb des Zustands "Ressourcen Verwaltung geöffnet" ist der Zustand "Aktionen angezeigt". Dieser Zustand zeigt dem Mediathekar die verschiedenen verfügbaren Aktionen im Zusammenhang mit der Ressourcenverwaltung.

Vom Zustand "Aktionen angezeigt" aus können drei verschiedene Zustände erreicht werden:

- Übergang vom Zustand "Aktionen angezeigt" zum Zustand "Ressourcen hinzufügen" erfolgt, wenn die Bedingung "Ressourcen hinzufügen" ausgewählt wird. Dadurch öffnet sich die Seite "Ressourcen hinzufügen".
- Übergang vom Zustand "Aktionen angezeigt" zum Zustand "Ressourcen editieren" erfolgt, wenn die Bedingung "Ressourcen editieren" ausgewählt wird. Dadurch öffnet sich die Seite "Ressourcen editieren".
- Übergang vom Zustand "Aktionen angezeigt" zum Zustand "Ressourcen löschen" erfolgt, wenn die Bedingung "Ressourcen löschen" ausgewählt wird. Dadurch öffnet sich die Seite "Ressourcen löschen".

Das Zustandsdiagramm bietet dem Mediathekar einen visuellen Überblick über die verschiedenen Zustände und Übergänge, die während der Nutzung der Mediaverse-Plattform auftreten können. Es unterstützt den Mediathekar bei der effektiven Verwaltung von Ressourcen und der Durchführung der entsprechenden Aktionen.

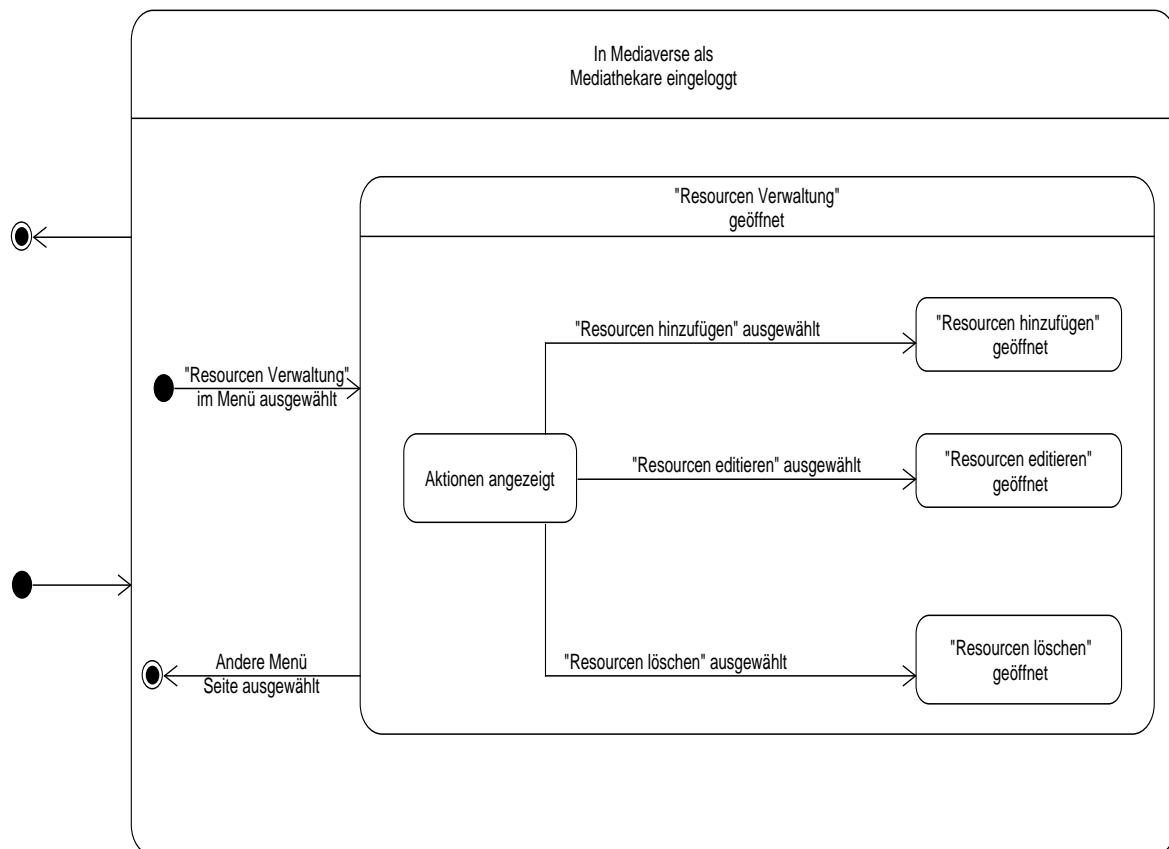


Abbildung 1.2: Statechart: Mediathekar

1.1.3 Admin

Der Statechart aus Abbildung 1.3 ergänzt den Statechart in Abbildung 1.1 für normale Benutzer, da der Administrator auch ein normaler Benutzer ist und somit über alle Funktionen eines normalen Benutzers verfügt. Dieses zusätzliche Statechart spezifiziert die Zustände und Übergänge, die speziell für Administratoren relevant sind.

Der Eingangszustand ist die "Admin-Seite geöffnet". Von diesem Zustand aus gibt es drei mögliche Zustandsübergänge, die von bestimmten Bedingungen abhängen.

Der Übergang zum Zustand "Benutzerseite angezeigt" erfolgt, wenn die Bedingung "Benutzer verwalten" im Menü ausgewählt wird. In diesem Zustand wird dem Admin die Benutzern Liste angezeigt, auf der er die Verwaltung der Benutzer durchführen kann. Zusätzlich hat der Administrator die Möglichkeit, neu registrierte Konten zu aktivieren, damit diese sich einloggen und auf das System zugreifen können.

Ein weiterer Übergang führt zum Zustand "Berichtsseite angezeigt", wenn die Bedingung "Berichte" im Menü ausgewählt wird. In diesem Zustand wird dem Admin die Berichtsseite angezeigt, auf der er Berichte einsehen kann.

Der dritte Übergang erfolgt, wenn die Bedingung "Ressourcen" im Menü ausgewählt wird, und führt zum Zustand "Ressourcenseite angezeigt". In diesem Zustand wird dem Admin die Seite angezeigt, auf der er auf Ressourcen zugreifen und diese verwalten bzw. neue Ressourcentypen hinzufügen kann.

Das Zustandsdiagramm bietet einen übersichtlichen Überblick über die verschiedenen Zustände und Übergänge, die ein Admin in Mediaverse durchläuft. Es ermöglicht dem Admin eine effektive Navigation zwischen den verschiedenen Seiten und die Ausführung seiner Aufgaben zur Verwaltung und Steuerung der Plattform.

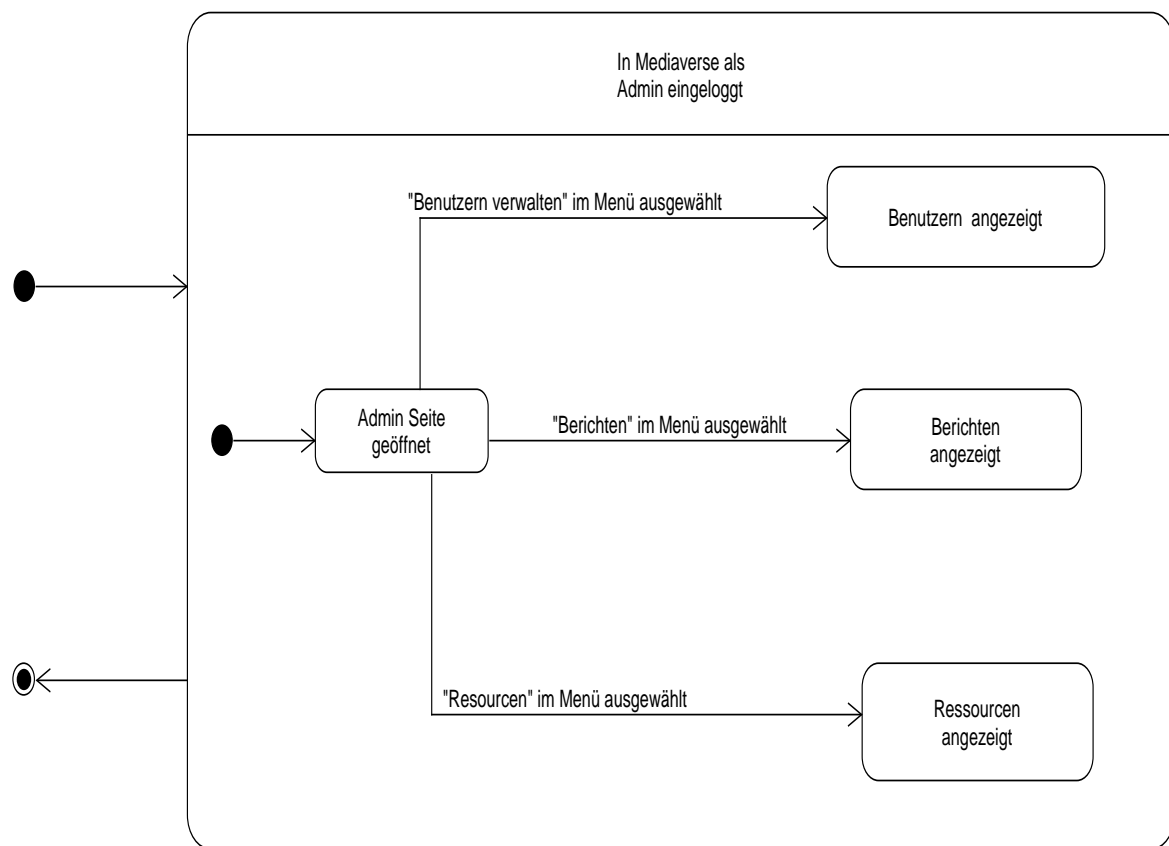


Abbildung 1.3: Statechart: Admin

1.2 Projektdetails

In diesem Abschnitt werden Prozesse, die durch das vorherige Statechart nicht ausreichend erläutert werden konnten, näher analysiert. Dafür werden Aktivitätsdiagramme genutzt, die den jeweiligen Systemabschnitt möglichst detailliert darstellen.

1.2.1 Verwaltung von Favoritenliste und teilen von Ressourcen

Hier wird ein Aktivitätsdiagramm 1.4 erstellt, um die verschiedenen Möglichkeiten der Benutzer zur Verwaltung und Freigabe von Favoritenlisten darzustellen. Das Diagramm zeigt die Schritte und Aktionen, die ein Benutzer ausführen kann, um Favoriten hinzuzufügen, zu löschen, zu bearbeiten und sie mit anderen Benutzern zu teilen. Der Prozess beginnt mit der Aktivität "Ressource suchen", bei der der Benutzer nach einer bestimmten Ressource innerhalb der Anwendung sucht. Nachdem er die gewünschte Ressource gefunden hat, kann er sie zur Favoritenliste hinzufügen, was die Aktivität "Zur Favoritenliste hinzufügen" darstellt.

Sobald Favoriten hinzugefügt wurden, kann der Benutzer die Favoritenliste anzeigen lassen und auf sie zugreifen, was durch die Aktivität "Favoritenliste anzeigen" repräsentiert wird, indem er auf "Favoritenliste" in der Menü klickt. Innerhalb der Favoritenliste kann der Benutzer eine Ressource auswählen, um weitere Aktionen durchzuführen. Diese Auswahl wird durch die Aktivität "Ressource auswählen" dargestellt.

Nachdem der Benutzer eine Ressource ausgewählt hat, gibt es zwei mögliche Aktivitäten, je nach den Bedürfnissen des Benutzers. Die Aktivität "Ressource anschauen" ermöglicht es dem Benutzer, die ausgewählte Ressource genauer zu betrachten und detaillierte Informationen darüber anzuzeigen. Alternativ kann der Benutzer die Aktivität "Ressource löschen" ausführen, um die Ressource aus seiner Favoritenliste zu entfernen. Wenn der Benutzer die Aktivität "Ressource anschauen" implementiert hat, besteht die Möglichkeit, die Ressource mit anderen Benutzern zu teilen. Diese Aktion wird durch die Aktivität "Ressource teilen" repräsentiert. Darüber hinaus, kann der Benutzer seine geteilte Liste und die geteilte Liste anderer Benutzer sehen, indem er im Menü auf die "geteilte Liste" klickt.

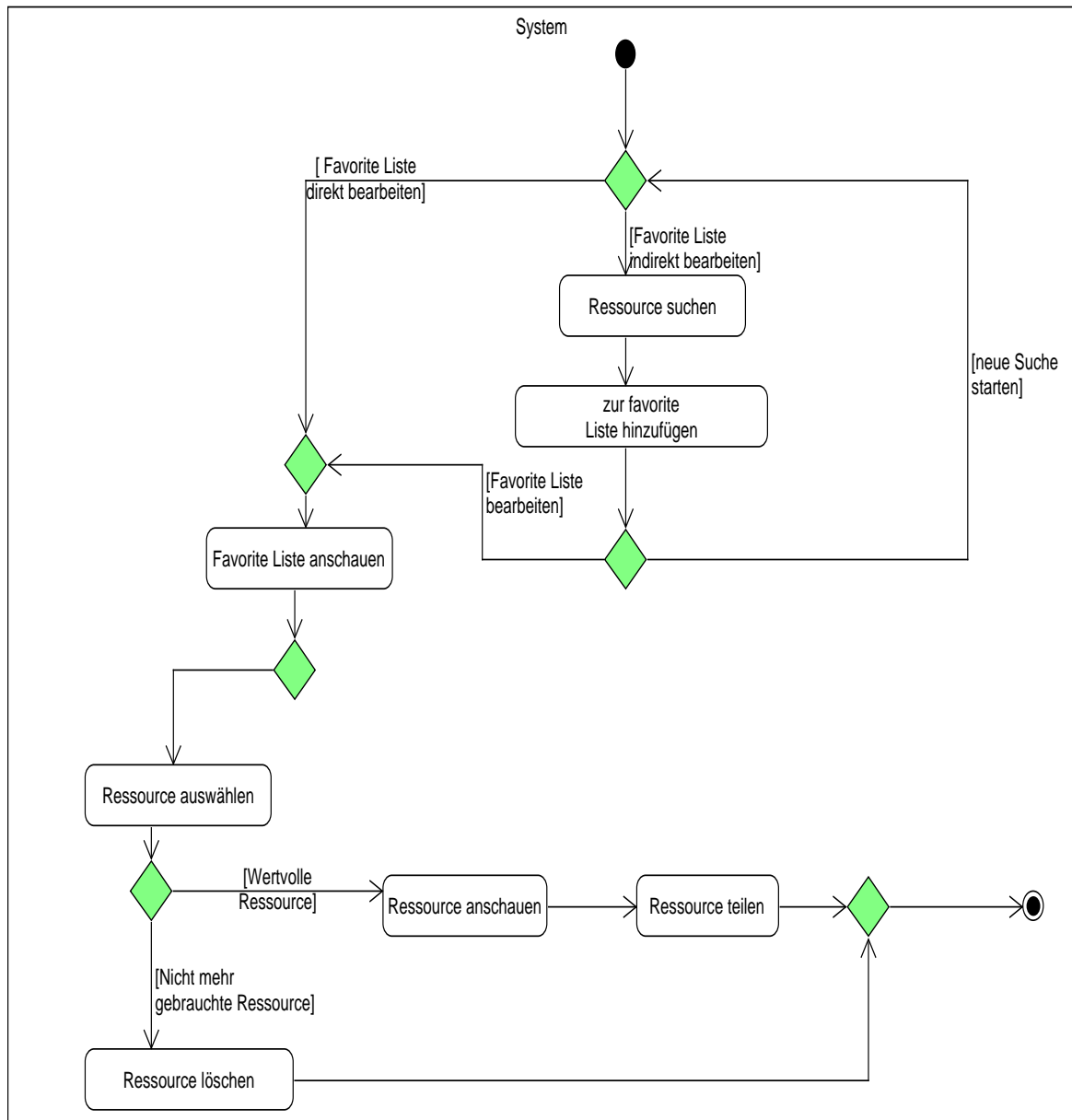


Abbildung 1.4: Aktivitätsdiagramm(UML): Favoritenliste

2 Analyse der Produktfunktionen

In diesem Kapitel wird das Verhalten für die einzelnen Produktfunktionen analysiert. Dies geschieht auf Basis der im Pflichtenheft analysierten Produktfunktionen und nicht-funktionalen Anforderungen, um später eine geeigneten Architektur realisieren zu können. Jede betrachtete Funktion wird in einem eigenen Unterkapitel dokumentiert.

2.1 Analyse von Funktionalität F10: Registrierung

Die Registrierungsseite einer Webanwendung ermöglicht Benutzern das Erstellen eines neuen Benutzerkontos. Der Ablauf beginnt, wenn der Benutzer die Registrierungsseite öffnet. Die Website lädt die erforderlichen Ressourcen und präsentiert dem Benutzer das Registrierungsformular.

Der Benutzer gibt seine erforderlichen Registrierungsinformationen ein, wie einen Benutzernamen, eine E-Mail-Adresse und ein Passwort.

Sobald der Benutzer alle erforderlichen Informationen eingegeben hat, sendet die Website diese Daten an den Webserver zur weiteren Verarbeitung.

Der Webserver empfängt die Registrierungsinformationen und überprüft sie auf ihre Gültigkeit. Dies kann verschiedene Überprüfungen beinhalten, z.B. ob der Benutzername bereits existiert oder ob das Passwort den Sicherheitsrichtlinien entspricht. Nachdem die Registrierung abgeschlossen ist, zeigt die Website dem Benutzer eine Erfolgsmeldung oder eine Fehlermeldung.

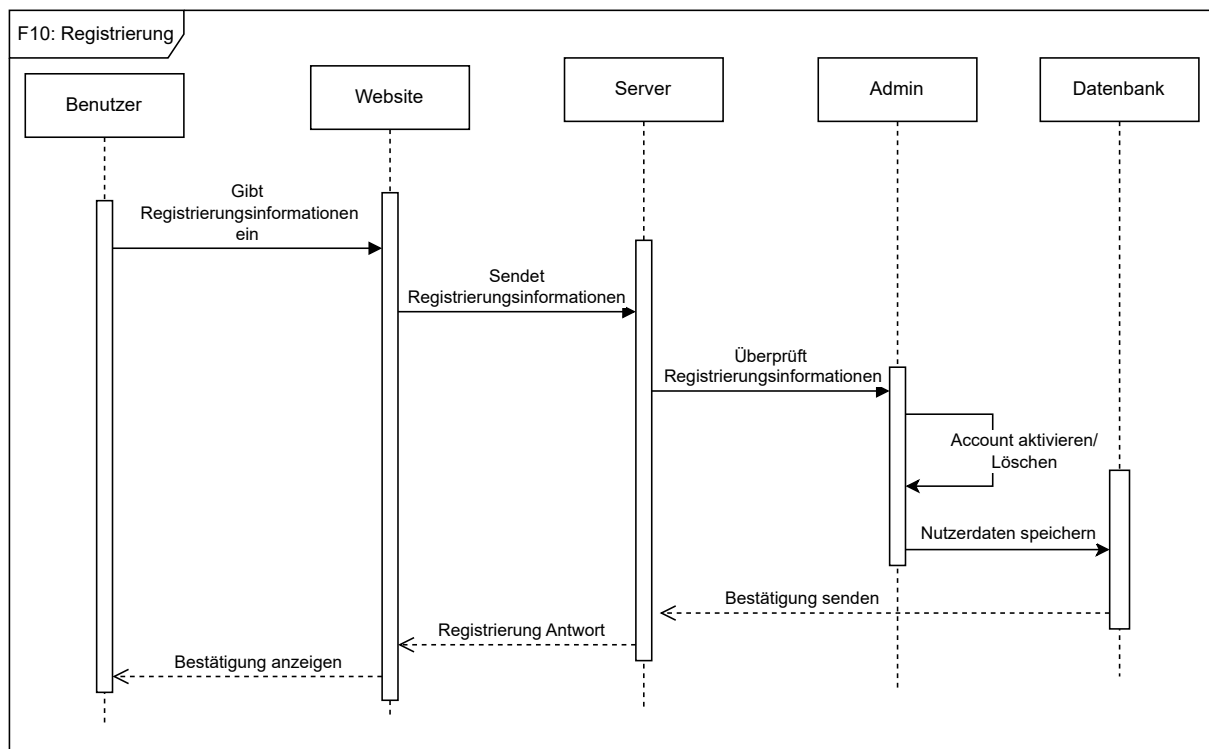


Abbildung 2.1: Sequenzdiagramm Registrierung

2.2 Analyse von Funktionalität F20: Login

Die Login-Seite einer Webanwendung ermöglicht Benutzern das Einloggen in ihr Benutzerkonto. Der Ablauf beginnt, wenn der Benutzer die Login-Seite öffnet. Die Website lädt die erforderlichen Ressourcen und präsentiert dem Benutzer die Login-Oberfläche. Dort gibt der Benutzer seine Anmeldeinformationen, Benutzername und Passwort, ein. Sobald der Benutzer seine Anmeldeinformationen eingegeben hat, sendet die Website diese Informationen an den Webserver.

Der Webserver empfängt die Anmeldeinformationen und leitet sie an die Datenbank weiter. Die Datenbank überprüft die Anmeldeinformationen auf ihre Gültigkeit, indem sie sie mit den gespeicherten Benutzerdaten vergleicht. Das Validierungsergebnis wird von der Datenbank an den Webserver zurückgesendet. Basierend auf dem Validierungsergebnis entscheidet die Website, ob der Benutzer erfolgreich eingeloggt wurde oder nicht.

Bei erfolgreichem Login zeigt die Website dem Benutzer eine Erfolgsmeldung an und leitet ihn zur Hauptseite oder einer anderen relevanten Seite der Webanwendung weiter. Dadurch kann der Benutzer nach dem Login in der Anwendung arbeiten oder navigieren. Wenn die Anmeldeinformationen ungültig sind oder ein Fehler aufgetreten ist, wird dem Benutzer eine Fehlermeldung angezeigt, die den Grund für das Scheitern des Logins erklärt.

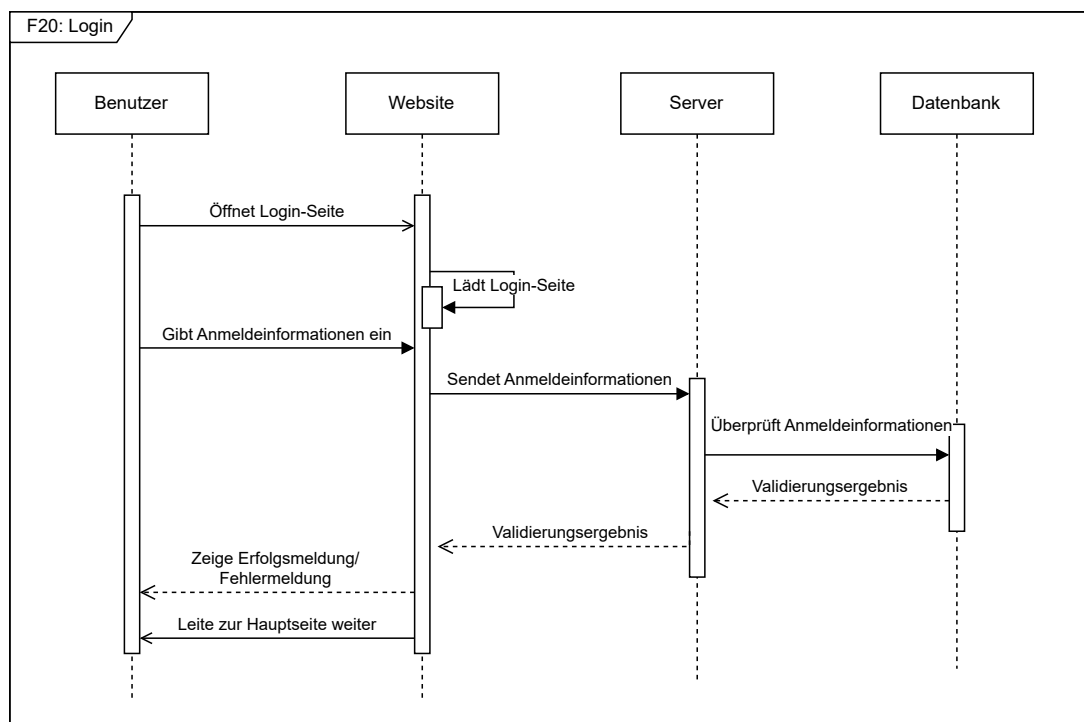


Abbildung 2.2: Sequenzdiagramm Login

2.3 Analyse von Funktionalität F30: Suchfunktion

Auf der Homepage kann der Benutzer in die Suchleiste einen Suchbegriff eingeben und auf den Button 'Suchen' klicken. Der Suchbegriff wird vom Browser an den Server übergeben und an die Datenbank weitergeleitet. Die Datenbank durchsucht nun die Daten auf Übereinstimmungen und gibt diese wieder an den Server zurück. Der Server leitet wieder die Treffer an den Browser zurück und dieser zeigt die Treffer, als Liste dem Benutzer an.

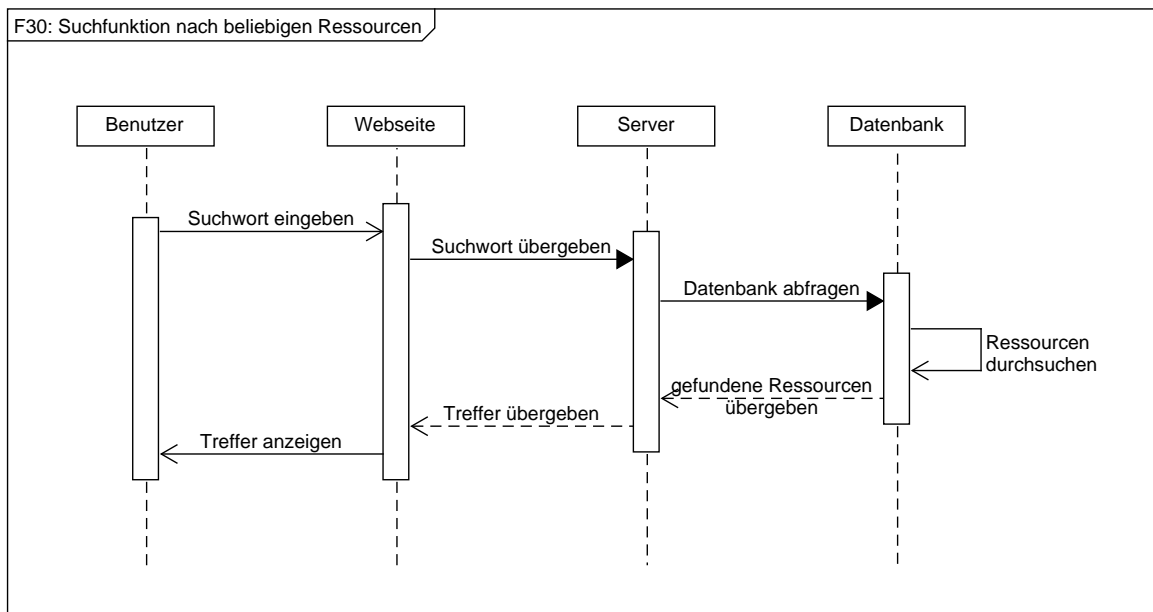


Abbildung 2.3: Sequenzdiagramm Suchfunktion

2.4 Analyse von Funktionalität F40: Verfügbarkeit, Ausleihen und Ausleih-Verlängerung

Hat sich der Benutzer für eine Ressource entschieden und möchte diese wenn möglich ausleihen, muss der Benutzer auf die Seite der Ressource gehen. Sobald er auf die Ressource geklickt hat empfängt der Server eine Verfügbarkeitsanfrage vom Browser. Der Server verarbeitet die Anfrage und stellt eine SQL-Anweisung an die Datenbank, ob die Ressource zur Ausleihe verfügbar ist. Danach verarbeitet die Datenbank diese Anweisung und gibt ein entsprechendes Ergebnis an den Server zurück. Dieser sendet das Ergebnis an den Browser und der Browser stellt das Ergebnis dem Benutzer zur Verfügung.

Ist die Ressource verfügbar, kann der Benutzer die Ressource ausleihen, in dem er auf den entsprechenden Button klickt. Der Browser sendet nun eine Anfrage an den Server und dieser leitet die Anfrage an die Datenbank weiter. Die Datenbank sucht die Ressource und markiert sie für die nächsten 14 Tage als blockiert beziehungsweise nicht ausleihbar für andere Benutzer. Daraufhin schickt sie eine Bestätigung an den Server und dieser leitet diese an den Browser weiter. Der Benutzer bekommt danach die Bestätigung vom Browser angezeigt.

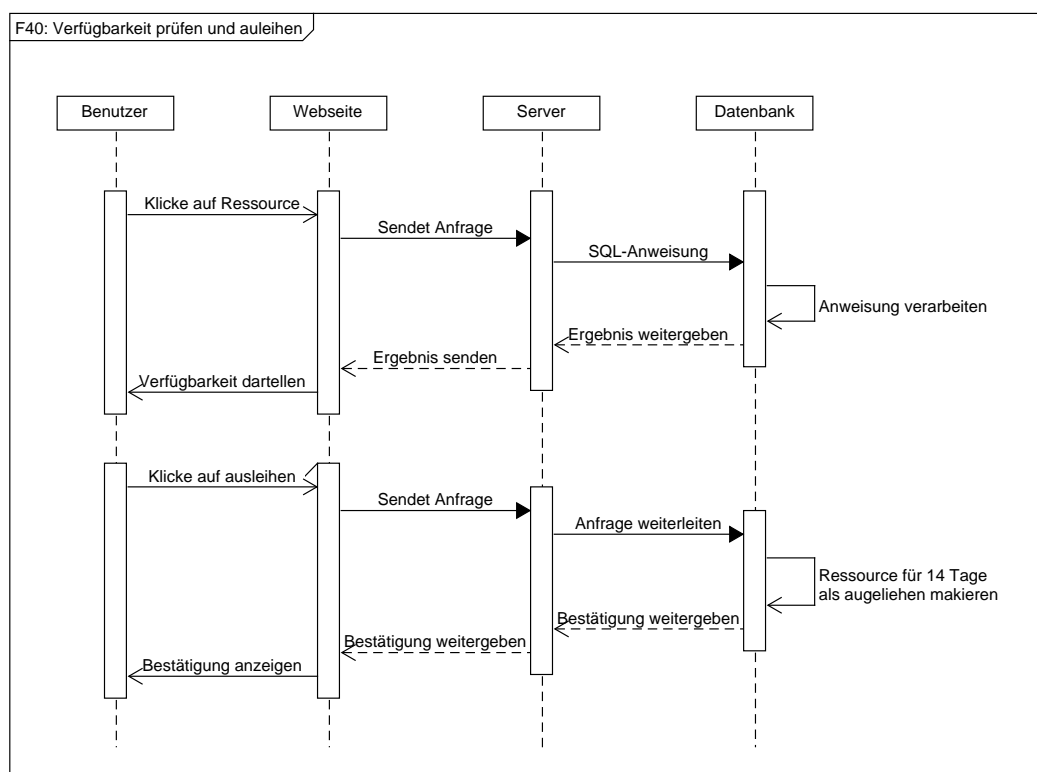


Abbildung 2.4: Sequenzdiagramm Verfügbarkeit und Ausleihen

Hat der Benutzer eine Ressource bereits ausgeliehen und möchte diese verlängern, muss er auf der Webseite auf '**Ausleihe verlängern**' klicken. Der Browser schickt nun die Anfrage an den Server und der leitet sie an die Datenbank weiter. Dort wird überprüft, ob alle Eingaben übereinstimmen. Wenn dies der Fall ist, wird die Ressource nochmals für weitere 14 Tage als ausgeliehen markiert und für andere Benutzer zur Ausleihe blockiert. Danach wird wieder eine Bestätigung an den Server zurück gesendet sowie auch vom Server an den Browser. Die Bestätigung bekommt der Benutzer wieder vom Browser angezeigt.

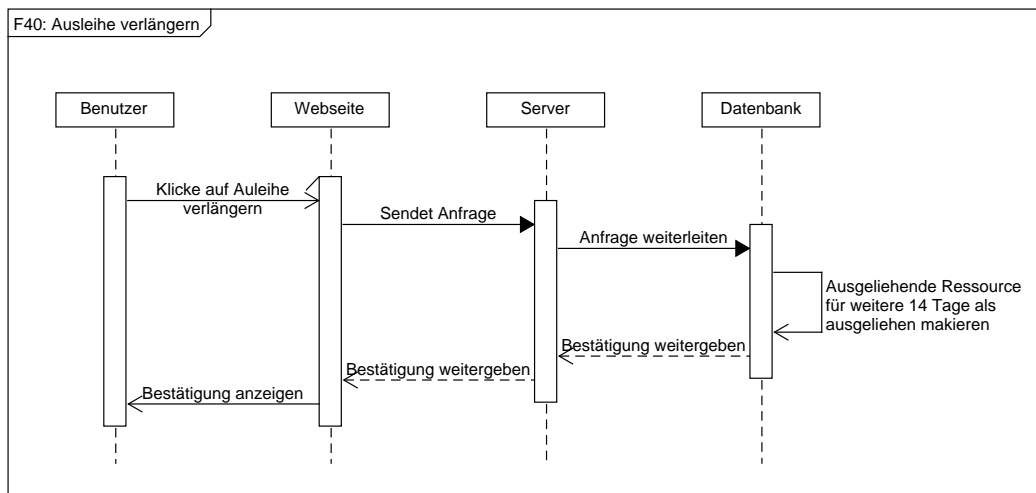


Abbildung 2.5: Sequenzdiagramm Ausleih-Verlängerung

2.5 Analyse von Funktionalität F50: Herunterladen

Der Benutzer ist auf der Seite einer herunter ladbaren Ressource und klickt auf den Button 'Download'. Daraufhin schickt der Browser dem Server eine Anfrage und dieser schickt eine Anweisung an die Datenbank. Stimmen alle Eingaben, ruft die Datenbank die Dateiinformationen auf und sendet die Datei an den Server zurück. Die Datei wird an den Browser weiter geleitet und dieser lädt nun die Datei herunter und speichert sie auf den Datenserver des jeweiligen Benutzers. Ist die Datei gespeichert worden, schickt der Datenserver dem Browser eine Bestätigung und zeigt die Datei dem Benutzer an.

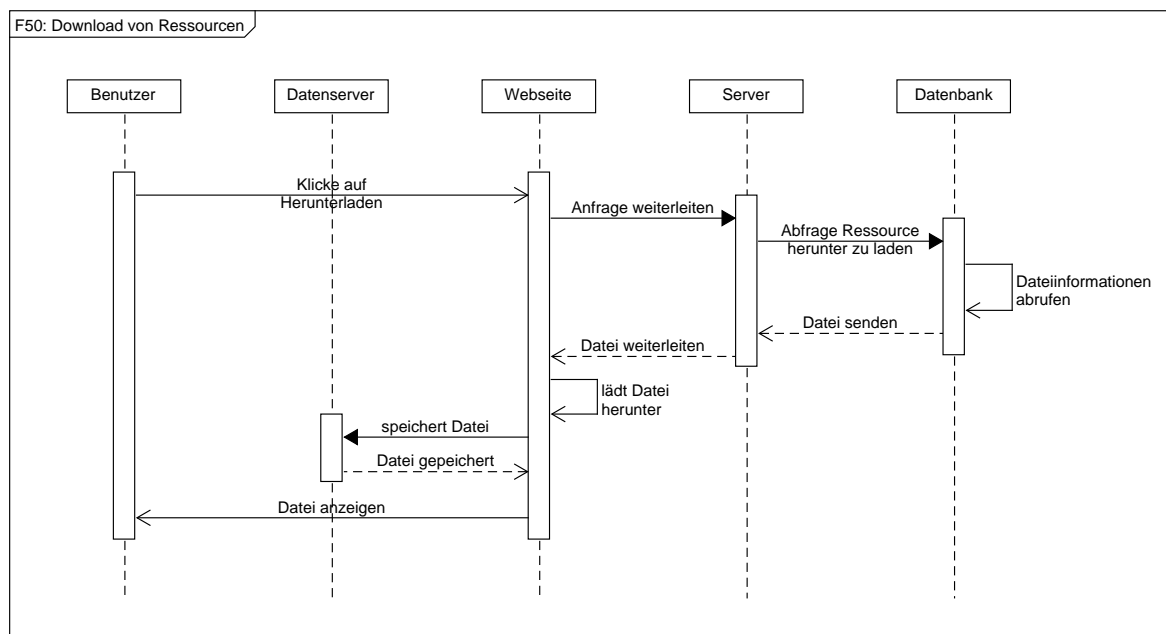


Abbildung 2.6: Sequenzdiagramm Download von Ressourcen

2.6 Analyse von Funktionalität F60: Hinzufügen, Bearbeiten, Löschen von Ressourcen

Wenn der Administrator eine neue Ressource hinzufügen möchte, gibt er der Website die erforderlichen Angaben. Die Website kommuniziert mit der Datenbank und speichert die Informationen über die Ressource. Sobald die Datenbank die erfolgreiche Speicherung bestätigt, benachrichtigt die Website den Administrator über das Hinzufügen der Ressource.

Um eine Ressource zu ändern, interagiert der Administrator mit der Website. Er teilt die spezifische Ressource und ihre aktualisierten Details mit. Die Website leitet die Anfrage an die Datenbank weiter, die die notwendigen Änderungen vornimmt.

Wenn der Administrator eine Ressource löschen möchte, soll er die betreffende Ressource auswählen und auf den Button '**Löschen**' klicken. Die Website leitet den Löschantrag an die Datenbank weiter, die die Ressource aus ihrem Speicher entfernt. Die Website bestätigt die Löschung gegenüber dem Administrator.

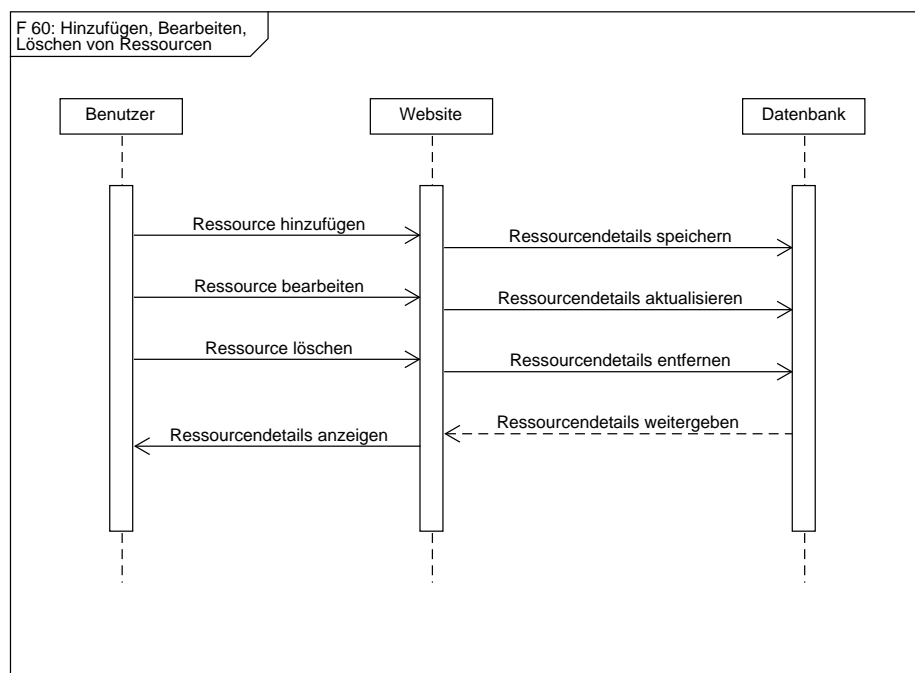


Abbildung 2.7: Sequenzdiagramm Hinzufügen, Bearbeiten, Löschen von Ressourcen

2.7 Analyse von Funktionalität F70: Favoritenlisten

Wenn ein Benutzer eine Ressource auswählt, hat er die Möglichkeit, sie zu seiner Favoritenliste hinzuzufügen. Diese Funktion ermöglicht es dem Nutzer, die Ressource als Favorit zu markieren, um in Zukunft leichter darauf zugreifen zu können.

Die Nutzer können ihre Favoritenliste auch über den entsprechenden Bereich auf der Website einsehen. So können sie alle Ressourcen, die sie zu ihrer Favoritenliste hinzugefügt haben, an einem Ort sehen.

Wenn ein Nutzer eine bestimmte Ressource nicht mehr in seiner Favoritenliste haben möchte, kann er sie löschen. Dadurch wird die Ressource aus der Liste entfernt und sichergestellt, dass die Favoritenliste aktuell bleibt und den Präferenzen des Nutzers entspricht.

Schließlich können Nutzern ihre Favoritenliste mit anderen teilen, indem sie auf den Button 'Teilen'.

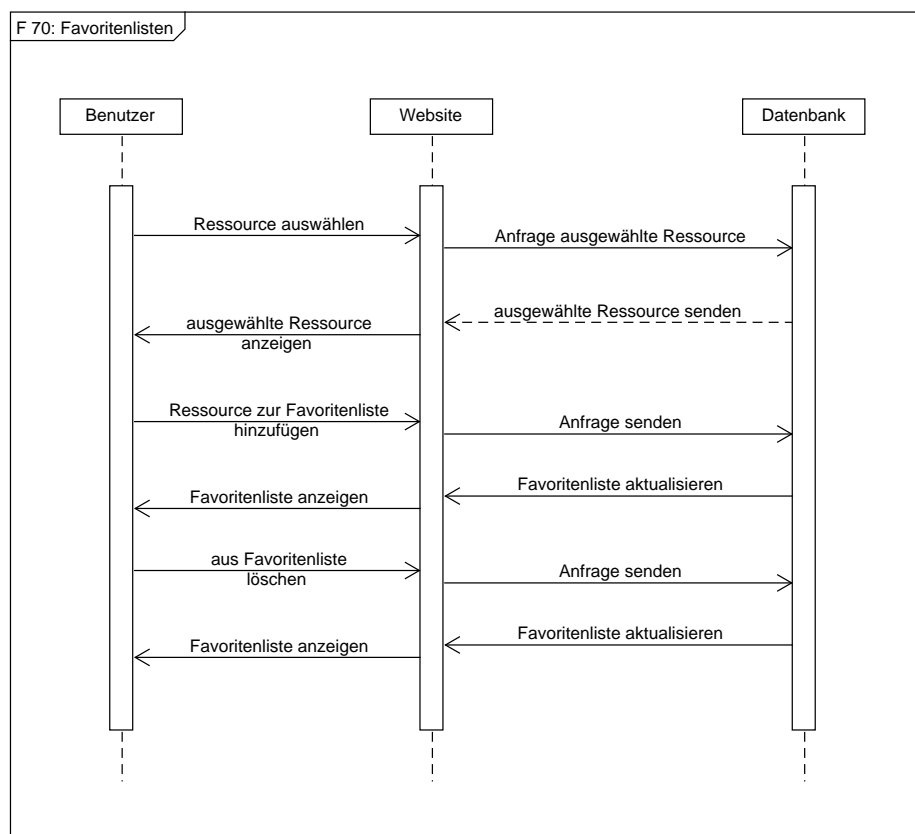


Abbildung 2.8: Sequenzdiagramm Favoritenlisten

2.8 Analyse von Funktionalität F80: Spracheinstellung

Die Spracheinstellungsseite ist eine Benutzeroberfläche, die es dem Benutzer ermöglicht, die bevorzugte Sprache für die Website auszuwählen. Die Seite wird aufgerufen, wenn der Benutzer auf den '**Help**' -Button klickt und anschließend die Option '**Change language**' auswählt.

Dann wird dem Benutzer eine Liste von verfügbaren Sprachoptionen präsentiert. Diese Liste wird als Dropdown-Menü dargestellt. Der Benutzer kann eine Sprache auswählen, indem er auf die entsprechende Option klickt.

Nachdem der Benutzer eine Sprache ausgewählt hat, wird die Auswahl erfasst. Dies wird automatisch durch eine Echtzeitaktualisierung geschehen. Nachdem die Benutzeroberfläche aktualisiert wurde, wird dem Benutzer die neue Ansicht präsentiert, in der die Inhalte in der ausgewählten Sprache angezeigt werden. Der Benutzer kann nun die Website in einer für ihn leichter verständlichen Sprache nutzen.

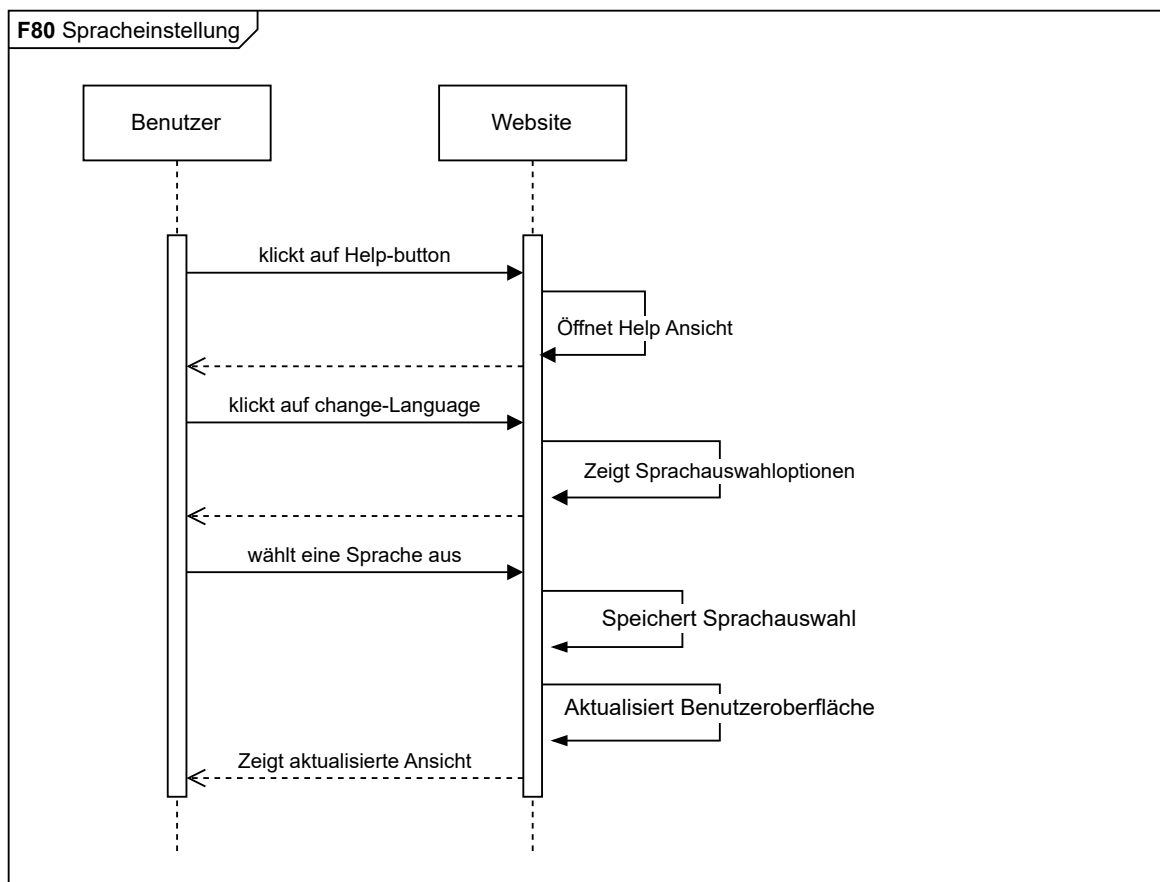


Abbildung 2.9: Sequenzdiagramm Spracheinstellung

2.9 Analyse von Funktionalität F90: Verlauf der (eigenen) Aktionen

Durch die Auswahl dieser Option erhalten die Nutzer Einblicke in ihre vergangenen Ausleihen und können die Dauer der Ausleihe jeder einzelnen Ressource einsehen.

Wenn ein Nutzer auf den Button '**Verlauf anzeigen**' klickt, wird die Anfrage vom Nutzer an die Website gesendet. Die Website empfängt die Anfrage und leitet eine Abfrage an die Datenbank ein, um die Ausleihhistorie des Benutzers abzurufen.

Die Website, die als Vermittler fungiert, ruft die Informationen zum Ausleihverhalten des Nutzers aus der Datenbank ab. Diese Daten enthalten Einzelheiten zu den Ressourcen, die der Nutzer in der Vergangenheit ausgeliehen hat, sowie die entsprechenden Ausleihzeiten.

Sobald die Website die Daten zum Ausleihverhalten erhalten hat, werden sie dem Nutzer in einem benutzerfreundlichen Format präsentiert. Der Nutzer kann dann seine Ausleihhistorie, einschließlich der Dauer der einzelnen Ausleihperioden, direkt auf der Website einsehen.

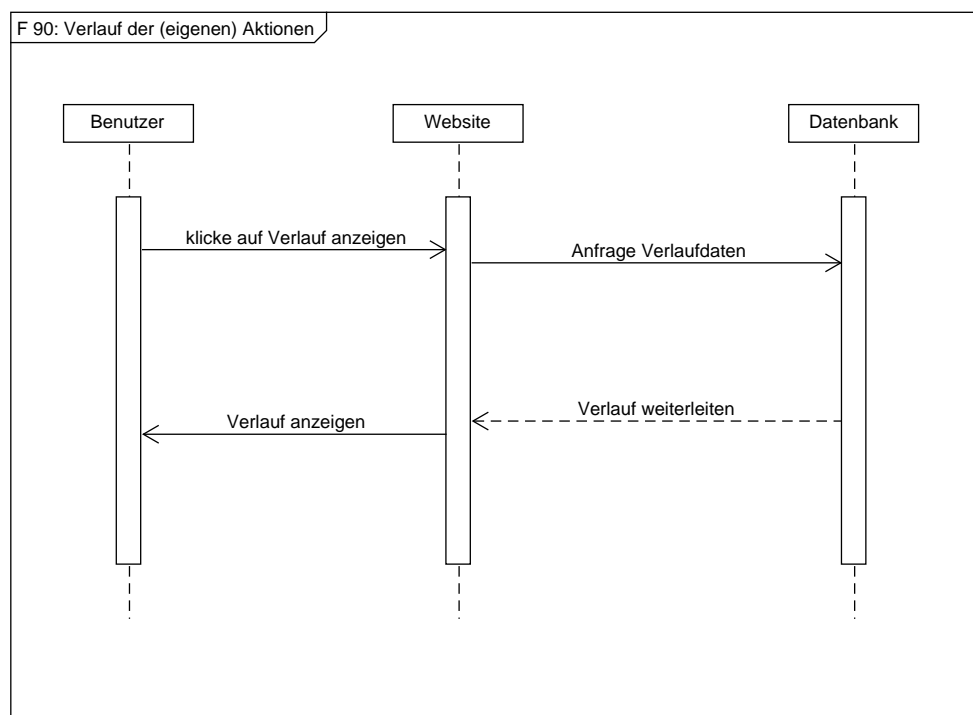


Abbildung 2.10: Sequenzdiagramm Verlauf der (eigenen) Aktionen

2.10 Analyse von Funktionalität F100: Anlegen von Benutzern und Mediathekaren

Mit dieser Funktion kann der Verwalter ein Konto entweder einem Benutzer oder einem Mediathekar hinzufügen.

Der Vorgang beginnt damit, dass der Verwalter das Konto erstellt. Der Verwalter sendet eine Anfrage an das System, um die Verfügbarkeit zu prüfen. Das System prüft, ob die gewünschten Kontodaten vorhanden sind oder nicht. Das System teilt dann das Ergebnis mit, indem es eine Antwort an den Verwalter zurücksendet. Nach Erhalt des Verfügbarkeitsergebnisses fährt der Verwalter mit der Validierung der Eingabe fort. Das System validiert die Eingabe und sendet eine Antwort an den Verwalter zurück.

Der Verwalter bestimmt die Berechtigung für das Konto (Benutzer oder Mediathekar). Der Verwalter sendet eine Anfrage an das System, um die Daten zu aktualisieren. Das System aktualisiert die Kontoberechtigung in der Datenbank und sendet eine Antwort an den Verwalter zurück.

Abschließend bestätigt das System den Verwalter darüber, dass das Konto erfolgreich erstellt wurde.

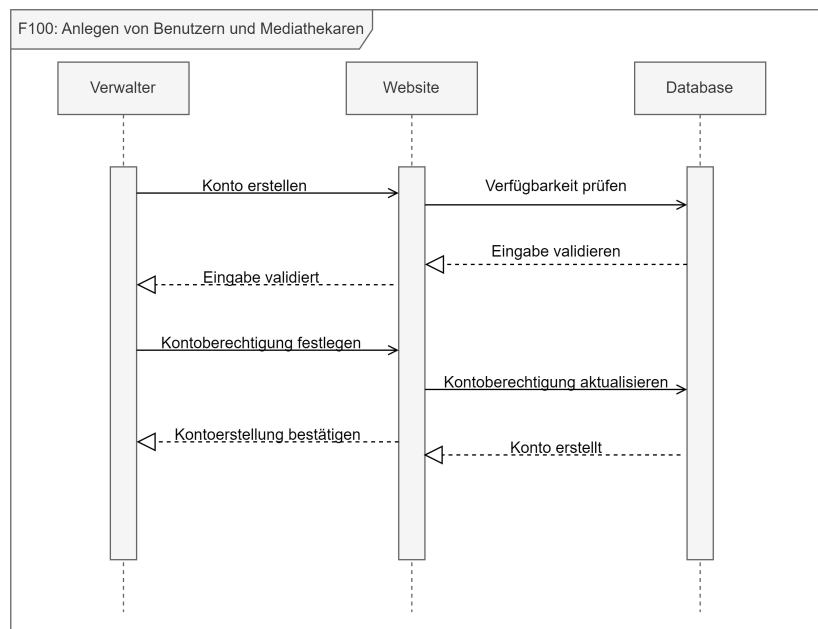


Abbildung 2.11: Sequenzdiagramm Anlegen von Benutzern und Mediathekaren

2.11 Analyse von Funktionalität F110: Editieren und Löschen von Benutzern und Mediathekaren

Der Administrator ist in der Lage, Konten zu verwalten und zu editieren, z.B. seine Berechtigung zu ändern (von Benutzer zu Mediathekar oder umgekehrt) oder das Konto vollständig zu löschen.

Der Vorgang beginnt damit, dass der Administrator das Konto verwaltet. Das System ruft die Liste der Konten aus der Datenbank ab und sendet sie an den Administrator. Nach Erhalt der Kontoliste wählt der Administrator ein bestimmtes Konto zur Verwaltung aus. Der Administrator wählt dann entweder die Kontoberechtigung zu verändern oder das Konto zu löschen. Wenn der Administrator die Kontoberechtigung ändern möchte, benachrichtigt das System den Administrator darüber, dass die Kontoberechtigung erfolgreich geändert wurde. Wenn der Administrator das Konto löschen möchte, entfernt das System das Konto aus der Datenbank und benachrichtigt den Administrator, dass das Konto erfolgreich entfernt wurde.

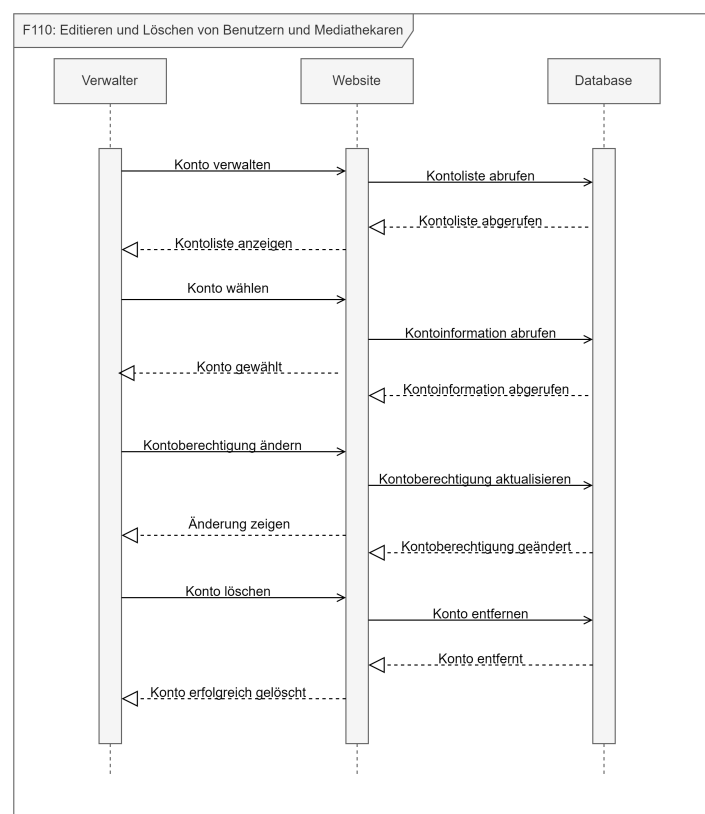


Abbildung 2.12: Sequenzdiagramm Editieren und Löschen von Benutzern und Mediathekaren

2.12 Analyse von Funktionalität F120: QR-Code

Neben der Suchfunktion können Benutzer auch über den QR-Code nach Ressourcen suchen.

Der Ablauf beginnt mit der erfolgreichen Anmeldung des Benutzers am System. Der Benutzer klickt auf 'QR-Code', um den QR-Code-Scanvorgang zu starten. Das System leitet den Benutzer zur gerätespezifischen Kameraschnittstelle weiter, um ein Foto aufzunehmen. Der Benutzer fügt über die Gerätekamera ein Foto des QR-Codes ein. Das System verarbeitet den QR-Code, indem es die relevanten Informationen extrahiert. Das System validiert den QR-Code, um seine Authentizität und Integrität sicherzustellen. Nach erfolgreicher Validierung benachrichtigt das System den Benutzer, dass die gewünschte Ressource gefunden wurde. Das System fährt fort, dem Benutzer die Ressourceninformationen anzuzeigen.

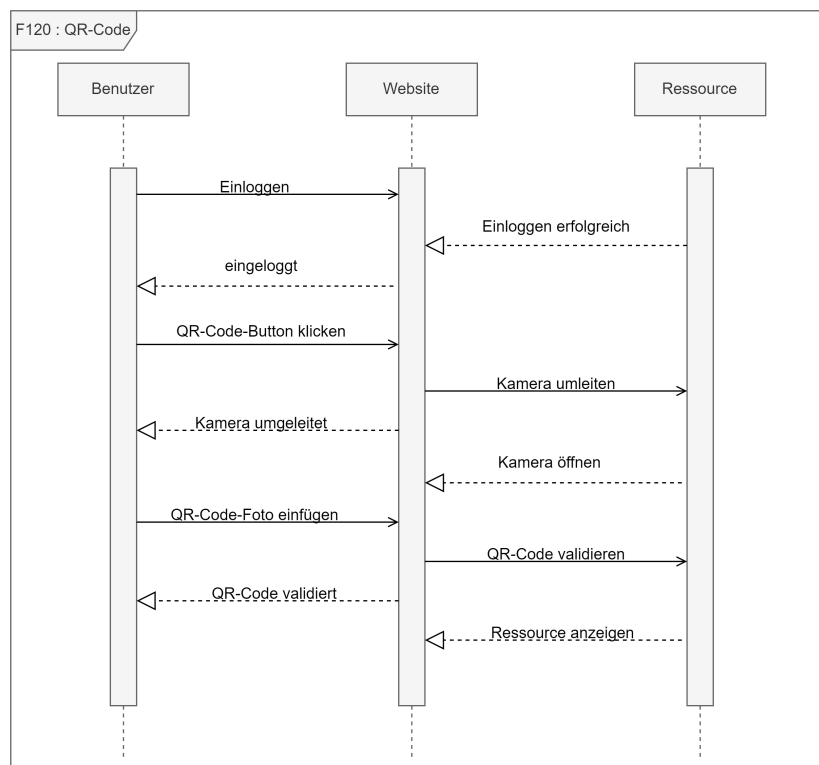


Abbildung 2.13: Sequenzdiagramm QR-Code

3 Resultierende Softwarearchitektur

Dieser Abschnitt hat die Aufgabe, einen Überblick über die zu entwickelnden Komponenten und Subsysteme zu liefern.

3.1 Komponentenspezifikation

In diesem Abschnitt wird die aus der Analyse der Produktfunktionen (Kapitel 2) resultierende Komponentenstruktur durch der Komponentendiagramm in Abbildung 3.1 beschrieben.

Mediaverse ist in sieben größere Komponenten gegliedert. Es ist jedoch zu beachten, dass die Sprachverwaltungskomponente nicht im Komponentendiagramm enthalten ist, da sie eine systemweite Einstellung darstellt, die für das gesamte System gilt.

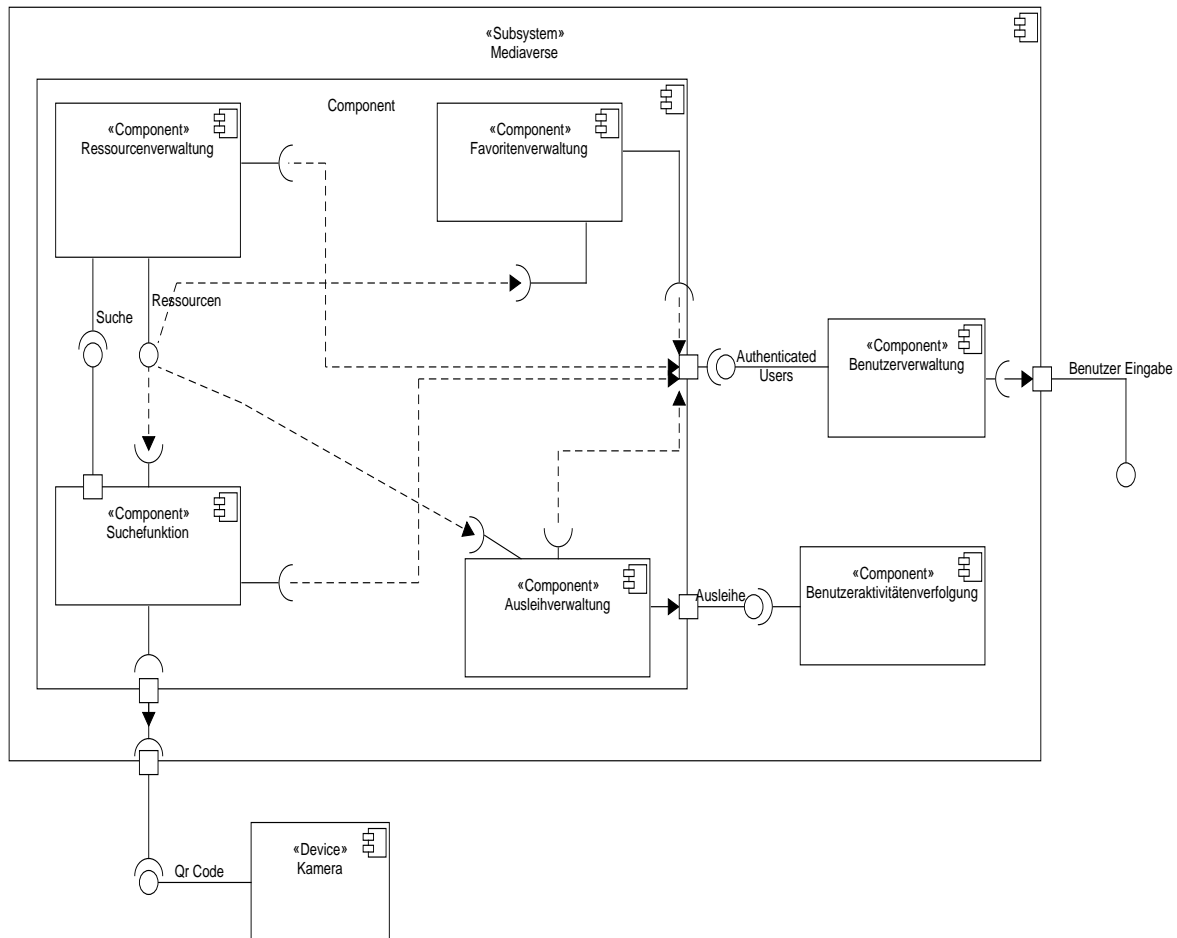


Abbildung 3.1: Komponentendiagramm

Nachfolgend werden die Aufgaben der einzelnen Komponenten kurz beschrieben.

Komponente $\langle C10 \rangle$: Benutzerverwaltung

Die Benutzerverwaltungskomponente ist für die Verwaltung verschiedener Aufgaben rund um Benutzer zuständig. Dazu gehört die Registrierung neuer Benutzer sowie die Verwaltung von Benutzerkonten und dazugehörigen Informationen. Die Komponente kümmert sich um die Authentifizierung von Benutzern und stellt sicher, dass der Zugriff auf das System sicher und kontrolliert erfolgt.

Die Funktionalität der Benutzerverwaltungskomponente variiert leicht je nach Benutzertyp.

Komponente $\langle C20 \rangle$: Ressourcenverwaltung

Die Ressourcenverwaltungskomponente spielt eine zentrale Rolle bei der effizienten Verwaltung von Medienressourcen. Je nach Benutzertyp bietet sie umfangreiche Funktionen zum Hinzufügen, Bearbeiten und Löschen von Ressourcen.

Benutzer können mithilfe dieser Komponente neue Ressourcen erstellen und vorhandene Ressourcen entsprechend ihrer Berechtigungen anpassen oder entfernen. Zusätzlich besteht die Möglichkeit, die Ressourcen in verschiedene Kategorien einzuteilen, um eine geordnete Struktur und eine einfache Auffindbarkeit zu gewährleisten.

Darüber hinaus können Benutzer über diese Komponente auf die Ressourcen zugreifen, um weitere Informationen darüber zu erhalten oder sie herunterzuladen.

Komponente $\langle C30 \rangle$: Suchfunktion

Die Suchfunktion ist eine wichtige Komponente, die Benutzern ermöglicht, gezielt nach Medienressourcen zu suchen. Sie bietet Filter- und Sortieroptionen für präzise Ergebnisse und eine benutzerfreundliche Darstellung der Suchergebnisse. Ein weiteres hilfreiches Feature ist die Integration von QR-Codes, die die Suche erleichtern und eine schnellere Navigation zu bestimmten Ressourcen ermöglichen.

Komponente $\langle C40 \rangle$: Ausleihverwaltung

Die Ausleihverwaltungskomponente ist für die Verwaltung von Ausleihvorgängen zuständig. Benutzer können Medien ausleihen, diese zurückgeben und bei Bedarf die Ausleihfrist verlängern. Die Komponente überprüft die Verfügbarkeit der Medien für die Ausleihe und stellt sicher, dass sie korrekt verwaltet werden.

Komponente $\langle C50 \rangle$: Favoritenverwaltung

Die Favoritenverwaltungskomponente ermöglicht es Benutzern, ihre bevorzugten Ressourcen und Merklisten zu verwalten. Benutzer können Ressourcen zu ihren Favoriten hinzufügen, entfernen und bearbeiten, um schnell und einfach auf sie zugreifen zu können. Darüber hinaus bietet die Komponente die Möglichkeit, Elemente aus der Favoritenliste zu teilen, sodass andere Benutzer darauf zugreifen können. Dadurch können Benutzer ihre Empfehlungen und interessante Ressourcen mit anderen teilen und gemeinsam entdecken.

Komponente $\langle C60 \rangle$: Benutzeraktivitätenverfolgung

Die Benutzeraktivitätenverfolgungskomponente dient zur Aufzeichnung und Verfolgung von Benutzeraktivitäten und Verlauf. Dabei werden verschiedene Funktionen je nach Benutzertyp unterstützt.

Mediathekare und Administratoren können über diese Komponente einen Einblick in die von ihnen erstellten Ressourcen erhalten. Dadurch haben sie die Möglichkeit, den Überblick über ihre eigenen Beiträge und Aktivitäten zu behalten.

Benutzer können auch auf ihre eigene Aktivitätshistorie zugreifen und einsehen, welche Ressourcen sie in der Vergangenheit ausgeliehen oder genutzt haben. Dies ermöglicht es ihnen, ihren Verlauf nachzuvollziehen.

Komponente $\langle C70 \rangle$: Sprachverwaltung

Die Sprachverwaltungskomponente ist dafür verantwortlich, die Unterstützung mehrerer Sprachen und die Lokalisierung der Anwendung sicherzustellen. Sie bietet Funktionen zur Verwaltung sprachspezifischer Texte und Übersetzungen.

Durch die Sprachverwaltungskomponente können Benutzeroberfläche und Inhalte der Anwendung basierend auf der Spracheinstellung des Benutzers angepasst werden.

3.2 Schnittstellenspezifikation

Im Folgenden werden die einzelnen Schnittstellen der Komponenten aus der Komponentenspezifikation näher erläutert.

Schnittstelle $\langle I10 \rangle$: $\langle \text{Authenticated Users} \rangle$

Operation	Beschreibung
User registerUser(userData: UserData)	Ermöglicht die Registrierung eines neuen Benutzers. Sie erhält die Benutzerdaten als Parameter und fügt den Benutzer zur Benutzerdatenbank hinzu.
boolean activateUser(userId: Integer)	Ermöglicht es einem Administrator, einen neuen Benutzer zu aktivieren, damit dieser sich einloggen kann. Sie erhält die Benutzer-ID als Parameter und ändert den Status des Benutzers auf "aktiviert" in der Benutzerdatenbank.
User getUser(userId: Integer)	Liefert die Benutzerinformationen für einen bestimmten Benutzer anhand seiner username, email und User-type.
UpdatedUser updateUser(userId: Integer, userData: UserData)	Aktualisiert die Benutzerinformationen für einen bestimmten Benutzer anhand seiner Benutzer-ID.
string authenticateUser(username: string, password: string)	Authentifiziert einen Benutzer anhand seines Benutzernamens und Passworts. Sie gibt einen Token zurück, der verwendet werden kann, um die Authentifizierung des Benutzers bei nachfolgenden Anfragen zu überprüfen.

Schnittstelle $\langle I20 \rangle$: $\langle \text{Ressourcen} \rangle$

Operation	Beschreibung
Resource createResource(resourceData: ResourceData)	Ermöglicht es Mediathekaren und Administratoren, eine neue Ressource zu erstellen. Sie erhält die Ressourcendaten als Parameter und fügt die Ressource der Ressourcendatenbank hinzu. Die Operation gibt die neue Ressource zurück.
updatedResource updateResource(resourceId: Integer, updatedData: ResourceData)	Ermöglicht es Mediathekaren und Administratoren, eine vorhandene Ressource zu aktualisieren. Sie erhält die Ressourcen-ID und die aktualisierten Ressourcendaten als Parameter und aktualisiert die entsprechende Ressource in der Ressourcendatenbank.
void deleteResource(resourceId: Integer)	Ermöglicht es Mediathekaren und Administratoren, eine vorhandene Ressource zu löschen. Sie erhält die Ressourcen-ID als Parameter und entfernt die entsprechende Ressource aus der Ressourcendatenbank.
ResourceDetails getResourceDetails(resourceId: Integer)	Liefert die Details einer bestimmten Ressource anhand ihrer Ressourcen-ID. Sie gibt ein Objekt mit den relevanten Informationen der Ressource zurück

Schnittstelle $\langle I30 \rangle$: \langle Suche \rangle

Operation	Beschreibung
SearchResult[] search(query: string)	Ermöglicht es Benutzern, eine Suche basierend auf einem Suchbegriff durchzuführen. Sie erhält den Suchbegriff als Parameter und gibt eine Liste von Suchergebnissen zurück, die den Suchkriterien entsprechen.
SearchResult[] filter(criteria: FilterCriteria)	Filtiert die vorhandenen Suchergebnisse entsprechend der angegebenen Filterkriterien und liefert die gefilterte Liste zur weiteren Verarbeitung zurück.

Schnittstelle $\langle I40 \rangle$: \langle QR-Code \rangle

Operation	Beschreibung
Resource searchByQRCode(qrCode: string)	Ermöglicht es Benutzern, eine Ressource basierend auf dem QR-Code zu suchen. Sie erhält den QR-Code als Parameter und liefert die entsprechende Ressource zurück, sofern sie im System vorhanden ist.

Schnittstelle $\langle I50 \rangle$: \langle Ausleihe \rangle

Operation	Beschreibung
resource borrowResource(userId: Integer, resourceId: Integer)	Ermöglicht es einem Benutzer, eine Ressource auszuleihen. Sie erhält die Benutzer-ID und die Ressourcen-ID als Parameter und gibt die ausgeliehene Ressource zurück.
resource extendLoan(userId: Integer, loanId: Integer)	Ermöglicht es einem Benutzer, die Ausleihdauer einer Ressource zu verlängern. Sie erhält die Benutzer-ID und die Ausleih-ID als Parameter und gibt die aktualisierte Ressource zurück.
boolean checkAvailability(resourceId: Integer)	Überprüft die Verfügbarkeit einer Ressource. Sie erhält die Ressourcen-ID als Parameter und gibt einen booleschen Wert zurück, der angibt, ob die Ressource verfügbar ist oder nicht.
resource[] getLoanHistory(userId: Integer)	Liefert die Ausleihhistorie eines Benutzers. Sie erhält die Benutzer-ID als Parameter und liefert eine Liste von Ausleihen zurück, die der Benutzer in der Vergangenheit getätigt hat.
resource[] getCurrentBorrowedResources(userId: Integer)	Liefert die aktuell von einem Benutzer ausgeliehenen Ressourcen.

Schnittstelle $\langle I60 \rangle$: \langle Sprachverwaltung \rangle

Operation	Beschreibung
language[] show_lang_list()	Zeige unterstützte Sprachen
void switch_lang(HttpServletRequest request, str lang_code)	Wechsel zur ausgewählten Sprache

3.3 Protokolle für die Benutzung der Komponenten

Im folgenden Abschnitt wird für jede Komponente in Mediaverse festgehalten, wie diese korrekt verwendet werden soll. Dazu gibt es zu jeder Komponente einen Protokoll-Statechart, der den korrekten Umgang beschreibt, sowie einen erläuternden Text.

Es ist zu beachten, dass einige Komponenten möglicherweise keine direkten Berührungspunkte mit anderen Komponenten haben oder spezifisch für die Anforderungen von Mediaverse implementiert wurden, was ihre Wiederverwendbarkeit einschränkt, welche entsprechend in jedem Abschnitt erwähnt wird.

3.3.1 Protokoll für $\langle C10 \rangle$: Benutzerverwaltung

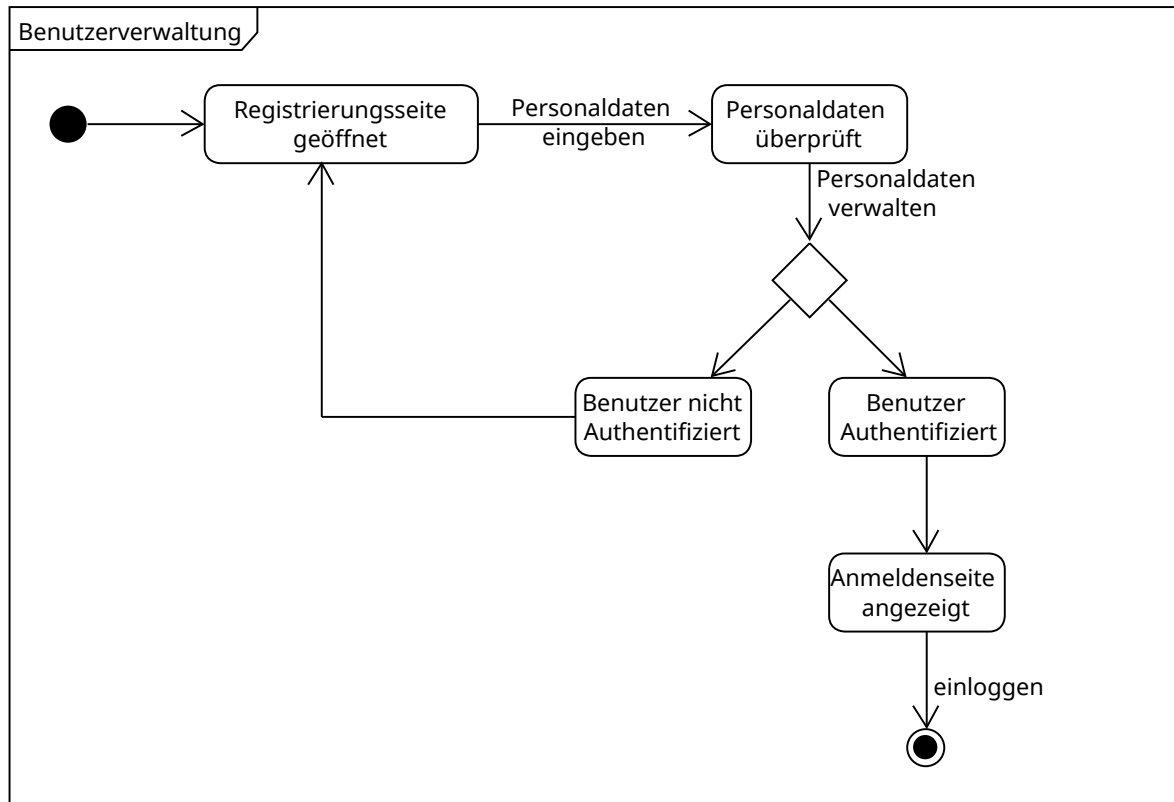


Abbildung 3.2: Zustandsdiagramm der Komponente $\langle C10 \rangle$

Die Benutzerverwaltungskomponente C10 stellt eine nahtlose Erfahrung für Benutzer bereit, die sich registrieren und auf das System zugreifen möchten. Zu Beginn öffnet der Benutzer die Registrierungsseite, wo er seine persönlichen Daten eingibt. Anschließend erfolgt eine sorgfältige Überprüfung der eingegebenen Informationen, um die Identität des Benutzers zu authentifizieren und sicherzustellen, dass nur legitime Benutzer Zugriff erhalten.

Nach erfolgreicher Authentifizierung wird der Benutzer automatisch zur Anmeldeseite weitergeleitet. Diese Seite ermöglicht es ihm, sich mit den entsprechenden Anmeldedaten einzuloggen und auf das System zuzugreifen. Die Benutzerverwaltungskomponente gewährleistet einen reibungslosen Übergang von der Registrierungsseite zur Anmeldeseite, wodurch ein komfortables und sicheres Benutzererlebnis gewährleistet wird.

3.3.2 Protokoll für $\langle C20 \rangle$: Ressourcenverwaltung

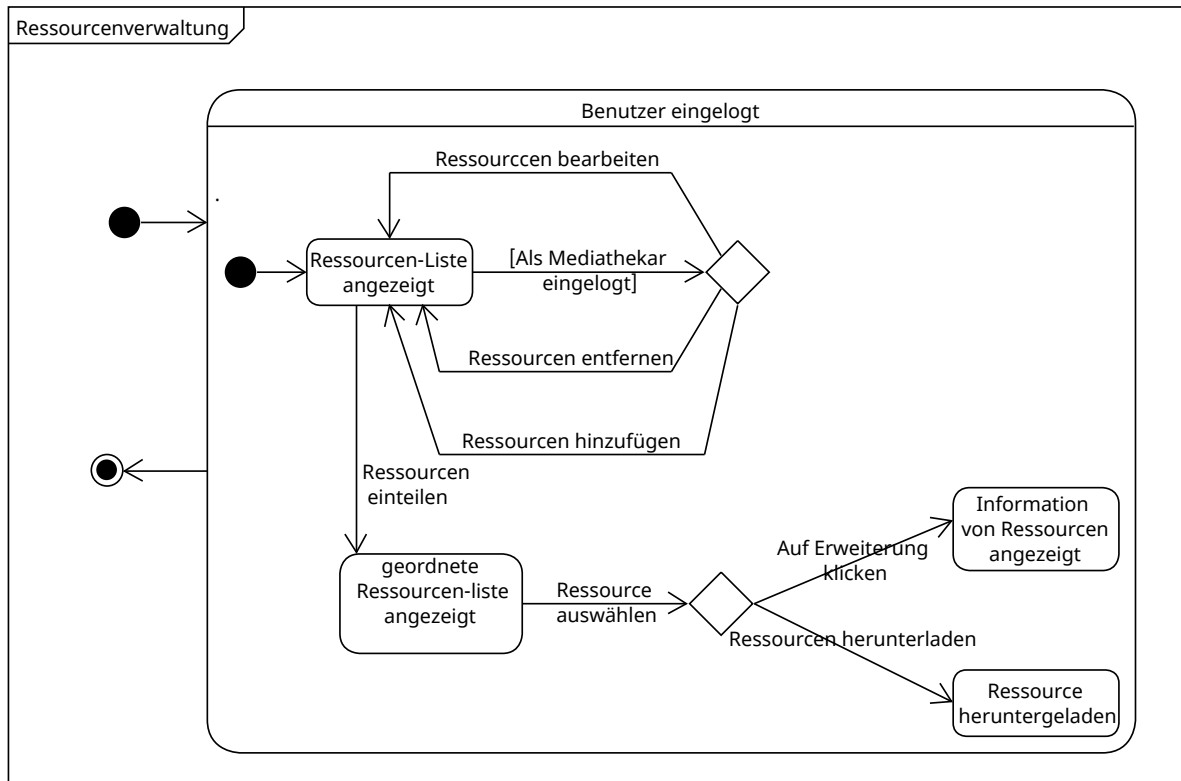


Abbildung 3.3: Zustandsdiagramm der Komponente $\langle C20 \rangle$

Die Ressourcenverwaltungskomponente C20 ist verantwortlich für die Verwaltung von Ressourcen. Mediathekare können mit Hilfe dieser Komponente neue Ressourcen hinzufügen, vorhandene Ressourcen bearbeiten oder löschen. Benutzer haben die Möglichkeit, die Ressourcenliste nach verschiedenen Kriterien zu ordnen und eine bestimmte Ressource auszuwählen, um deren Informationen einzusehen oder die Ressource herunterzuladen. Diese Komponente ermöglicht eine effiziente Verwaltung und Nutzung der verfügbaren Ressourcen in der Mediathek. Sie bietet eine benutzerfreundliche Schnittstelle, um Ressourcen hinzuzufügen, zu bearbeiten, zu löschen und zu durchsuchen, um eine reibungslose und vielseitige Nutzung der Ressourcen zu gewährleisten.

3.3.3 Protokoll für $\langle C30 \rangle$: Suchfunktion

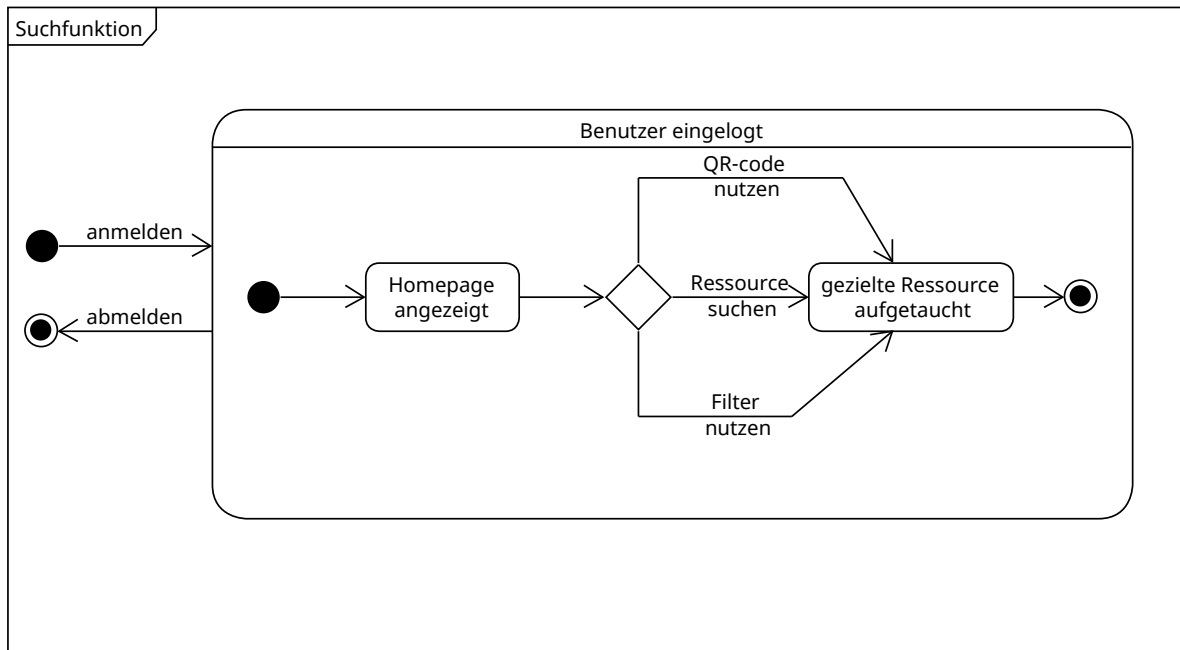


Abbildung 3.4: Zustandsdiagramm der Komponente $\langle C30 \rangle$

Die Suchfunktionskomponente ermöglicht Benutzern gezielte und präzise Suchanfragen für Medienressourcen. Mit einer benutzerfreundlichen Oberfläche und einer Vielzahl von Filter- und Sortieroptionen werden relevante Ergebnisse schnell angezeigt. QR-Code-Integration erleichtert die schnelle Navigation zu spezifischen Ressourcen. Die Suchfunktion verbessert die Effizienz und Benutzererfahrung bei der Suche nach Ressourcen im System.

3.3.4 Protokoll für $\langle C40 \rangle$: Ausleihverwaltung

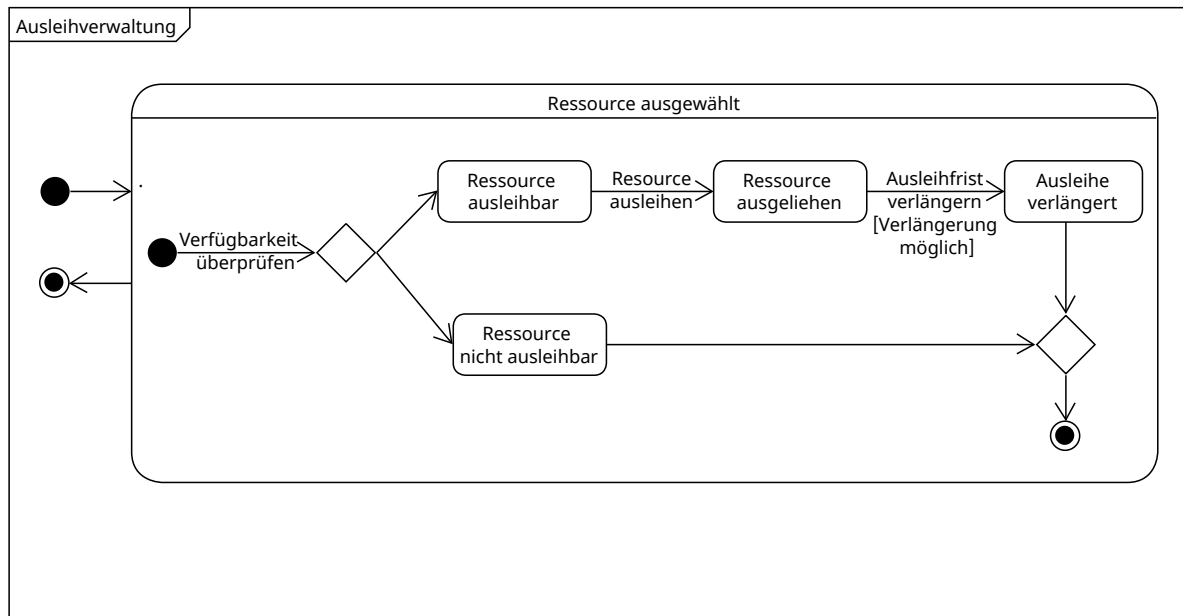
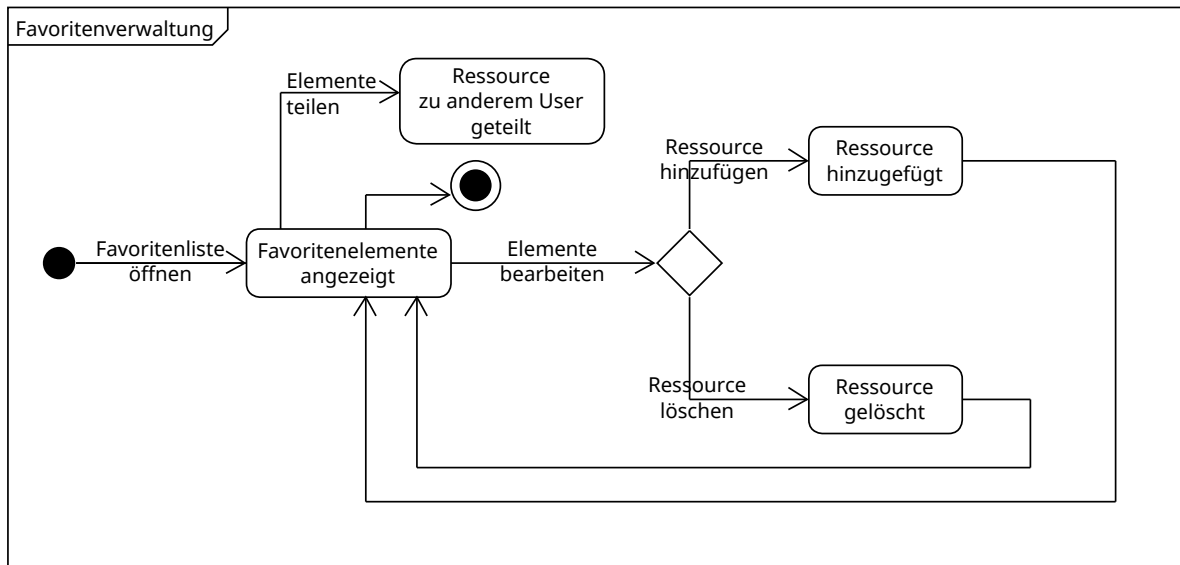


Abbildung 3.5: Zustandsdiagramm der Komponente $\langle C40 \rangle$

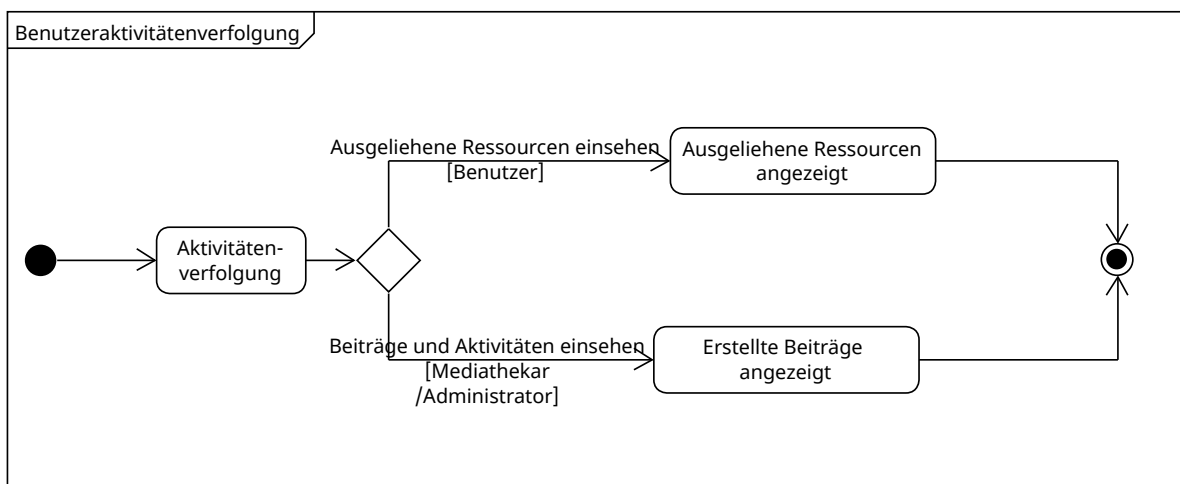
Die Ausleihverwaltungs-komponente $\langle C40 \rangle$ übernimmt die Überprüfung der Verfügbarkeit von Ressourcen für die Ausleihe, ermöglicht die Ausleihe von Ressourcen und bietet bei Bedarf die Möglichkeit, die Ausleihfrist zu verlängern. Diese Komponente gewährleistet eine effiziente Verwaltung von Ausleihvorgängen und stellt sicher, dass Benutzer problemlos auf die gewünschten Ressourcen zugreifen können. Durch die Überwachung der Verfügbarkeit, die Durchführung von Ausleihen und die Option zur Verlängerung der Ausleihfrist trägt die Komponente dazu bei, den Ausleihprozess reibungslos und kontrolliert abzuwickeln.

3.3.5 Protokoll für $\langle C50 \rangle$: Favoritenverwaltung

Abbildung 3.6: Zustandsdiagramm der Komponente $\langle C50 \rangle$

Die Favoritenverwaltungs-komponente verwaltet die Favoritenliste jedes Benutzers. Auf diese Weise können Benutzer ihre Favoritenliste bearbeiten, indem sie beliebige Ressourcen hinzufügen oder löschen. Benutzer haben außerdem die Funktion, ihre Favoritenliste mit anderen Benutzern zu teilen.

3.3.6 Protokoll für $\langle C60 \rangle$: Benutzeraktivitätenverfolgung

Abbildung 3.7: Zustandsdiagramm der Komponente $\langle C60 \rangle$

Die Benutzeraktivitätenverfolgungskomponente ermöglicht es jedem Benutzertyp, vom normalen Benutzern, Mediathekaren, bis zu Administratoren, den Verlauf seiner vergangenen Aktivitäten einzusehen. Mit dieser Komponente können Benutzer Ressourcen einsehen, die sie zuvor ausgeliehen haben, und Mediathekare und Administratoren können ihre vorherigen Uploads oder andere Aktivitäten einsehen.

3.3.7 Protokoll für $\langle C70 \rangle$: Sprachverwaltung

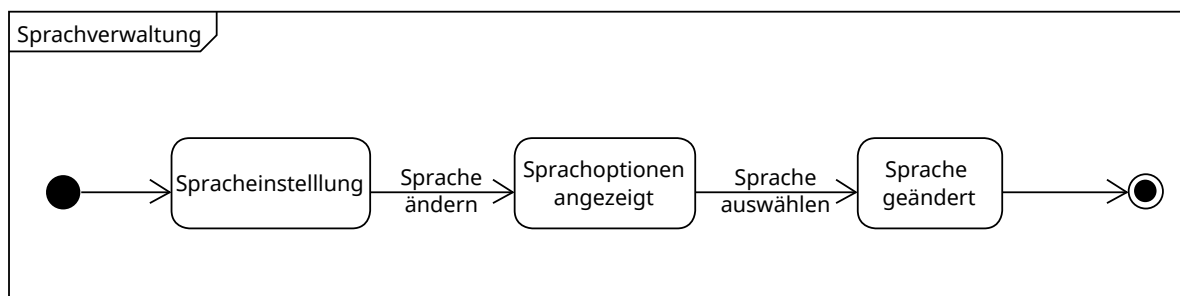


Abbildung 3.8: Zustandsdiagramm der Komponente $\langle C70 \rangle$

Die Sprachverwaltungskomponente bietet mehrsprachige Unterstützung der Applikation, wodurch die Benutzeroberfläche und der Inhalt der Applikation basierend auf den Sprachpräferenzen des Benutzers angepasst werden können.

4 Verteilungsentwurf

In Abbildung 4.1 wird die Anwendungsverteilung der Mediaverse dargestellt. Das System besteht aus verschiedenen Knotenkomponenten, die miteinander kommunizieren.

Der erste Knoten ist der Geräteknoten 'Server-PC', auf dem die Mediaverse-Anwendung gehostet wird. Der Server-PC enthält den Betriebssystemknoten 'Linux', der die Grundlage des Systems bildet und die Ausführung der Mediaverse-Anwendung ermöglicht. Zusätzlich ist die Komponente 'Datenbanksystem' im Server-PC enthalten, die für die Speicherung und Verwaltung der Mediaverse-Daten zuständig ist.

Im Betriebssystemknoten 'Linux' befinden sich zwei Umgebungsknoten: 'Python3' und 'JavaScript'. Diese Umgebungsknoten stellen die Laufzeitumgebungen für verschiedene Technologien dar. 'Python3' enthält das Framework 'Django', das das Teilsystem 'Mediaverse' umfasst. Dieses Teilsystem bildet den Kern der Anwendung und beinhaltet die wichtigsten Funktionen und Prozesse. Der Umgebungsknoten 'JavaScript' enthält das Framework 'Svelte', das ebenfalls das Teilsystem 'Mediaverse' beinhaltet. Durch die Verwendung verschiedener Umgebungsknoten können sowohl Python als auch JavaScript für die Entwicklung und Ausführung der Mediaverse-Anwendung genutzt werden.

Die Kommunikation zwischen den Umgebungsknoten erfolgt über definierte Schnittstellen. 'Python3' stellt APIs über das Framework 'Django Rest Framework' zur Verfügung, die von 'Svelte' mittels der 'fetch APIs' genutzt werden, um den Austausch von Daten zu ermöglichen. Zusätzlich gibt es eine Verbindung zwischen dem Betriebssystemknoten 'Linux' und dem Datenbanksystemknoten. Diese Verbindung wird durch einen 'DB Connector' realisiert. Der DB Connector stellt eine API bereit, die es dem Betriebssystem ermöglicht, auf die Datenbank zuzugreifen, Daten abzurufen und zu speichern.

Zusätzlich zur internen Kommunikation gibt es Verbindungen zwischen dem Server-PC, dem separaten Geräteknoten IRP-Server und dem separaten Geräteknoten 'Client-PC', der die Geräteknoten 'Kamera' enthält. Sowohl der Server-PC als auch der IRP-Server sind über den Kommunikationspfad 'HTTP' mit dem Client-PC verbunden. Der IRP-Server ist außerdem mit dem Server-PC über den Kommunikationspfad 'WSGI' verbunden. Dadurch wird der Austausch von Daten und die Interaktion zwischen den Komponenten ermöglicht.

Die Anwendungsverteilung der Mediaverse ermöglicht eine effiziente Kommunikation und Zusammenarbeit der Komponenten, um Konsistenz und einheitliche Kommunikationsmuster zu gewährleisten.

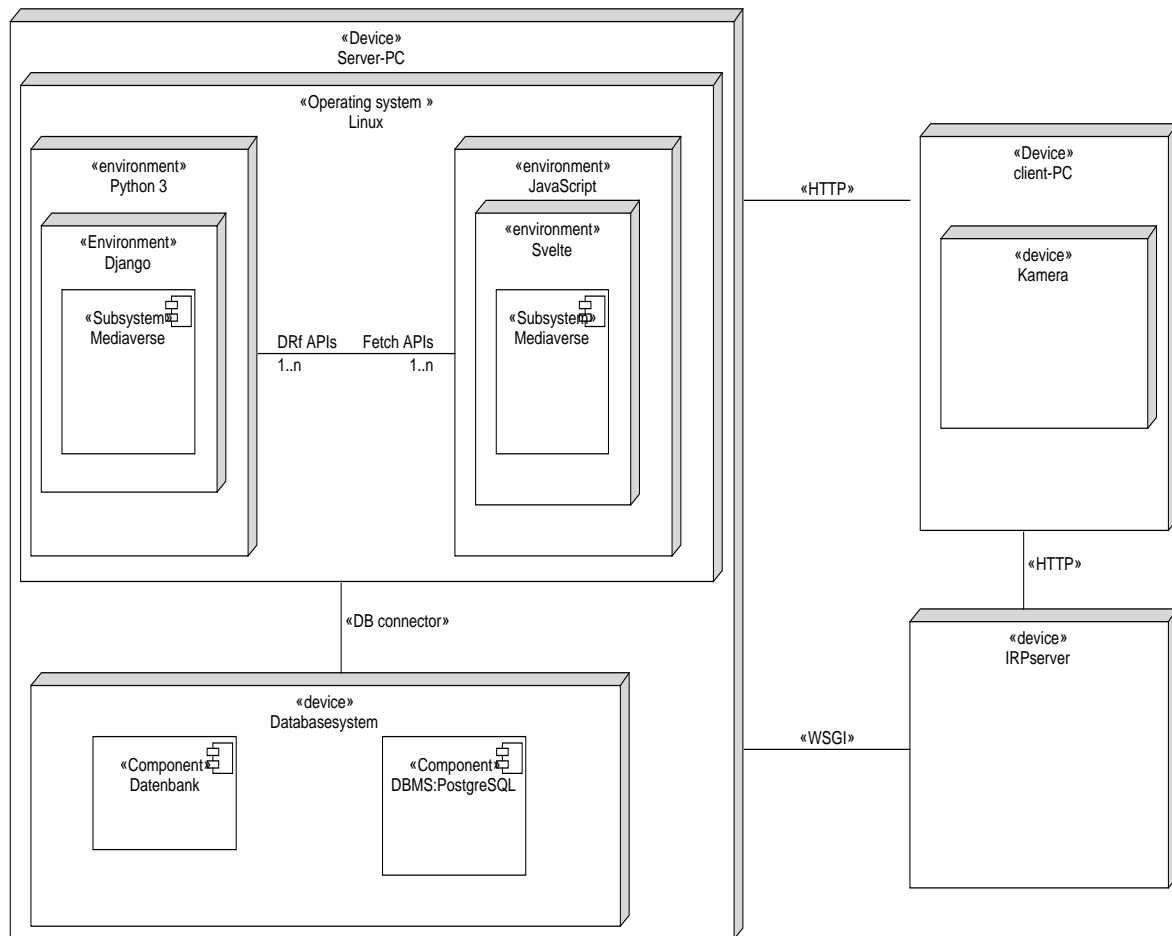


Abbildung 4.1: Verteilungsdiagramm

5 Implementierungsentwurf

Dieses Kapitel beschreibt die Implementierung der einzelnen Komponenten aus Kapitel 3. Dabei werden für jede Komponente die verwendeten Klassen, Bibliotheken und Attribute sowie Operationen näher erläutert und mithilfe von Klassendiagrammen erklärt.

5.1 Implementierung von Komponente $\langle C10 \rangle$: Benutzerverwaltung

Die Komponente $\langle C10 \rangle$ Benutzerverwaltung ist für die Verwaltung der gesamten Benutzer bzw. Benutzergruppen zuständig.

5.1.1 Paket-/Klassendiagramm

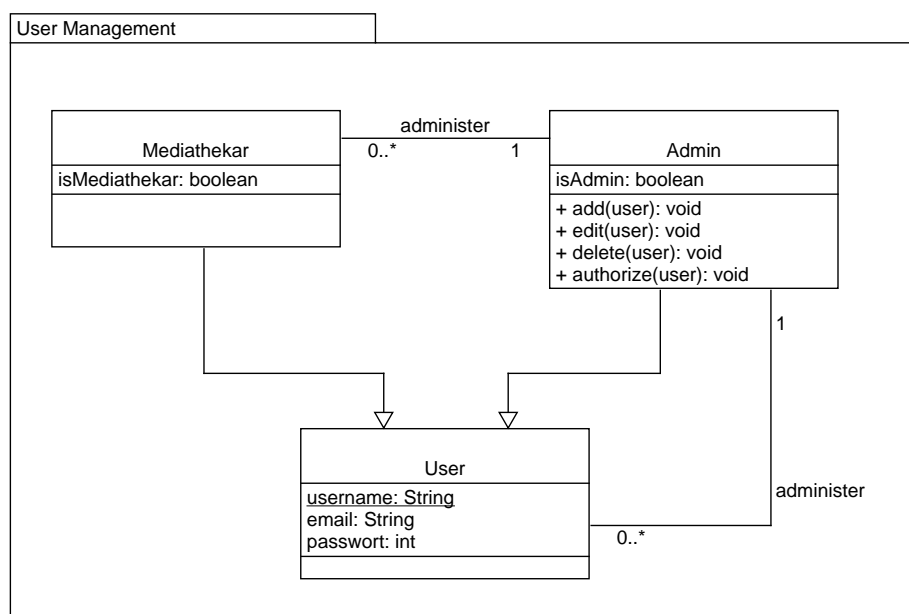


Abbildung 5.1: Klassendiagramm für Komponente $\langle C10 \rangle$

5.1.2 Erläuterung

Die Klasse Benutzer $\langle CL101 \rangle$ hat als Attribute einen eindeutigen Benutzernamen, eine E-Mail und ein Passwort, welche bei der Registrierung vergeben und gespeichert werden.

Die Klassen Mediathekar $\langle CL102 \rangle$ und Administrator $\langle CL103 \rangle$ erben beide vom Benutzer. Der Mediathekar und der Administrator haben jeweils noch Attribute zur Bestimmung der Rolle Mediathekar oder Administrator. Des Weiteren muss der Administrator neu registrierte Benutzer autorisieren und ihnen gegebenenfalls weitere Rechte vergeben, damit diese die Mediathek nutzen können. Außerdem kann er Benutzer hinzufügen, ändern und löschen. Dafür hat der Administrator die Operationen add, edit, delete und authorize.

Benutzer $\langle CL101 \rangle$

Aufgabe

Verwaltung der Benutzer-, Mediathekar- und Administratorkonten.

Attribute

username: Benutzername

email: E-Mail Adresse des jeweiligen Benutzers

password: Passwort zum Registrieren bzw. Anmelden

Operationen

Die Klasse hat keine weiteren Operationen.

Kommunikationspartner

$\langle CL103 \rangle$

Mediathekar $\langle CL102 \rangle$

Aufgabe

Verwaltung der Benutzer-, Mediathekar- und Administratorkonten.

Attribute

username: Benutzername

email: E-Mail Adresse des jeweiligen Benutzers

password: Passwort zum Registrieren bzw. Anmelden

Operationen

Die Klasse hat keine weiteren Operationen.

Kommunikationspartner

$\langle CL103 \rangle$

Administrator⟨CL103⟩

Aufgabe

Verwaltung der Benutzer-, Mediathekar- und Administratorkonten.

Attribute

username: Benutzername

email: E-Mail Adresse des jeweiligen Benutzers

password: Passwort zum Registrieren bzw. Anmelden

Operationen

add(user): Neue Benutzer beziehungsweise Mediathekare hinzufügen, die Rolle sagt aus, ob ein Benutzer oder ein Mediathekar hinzugefügt werden soll.

edit(user): Informationen zu einen bestehenden Benutzer, Mediathekar oder Administrator ändern.

delete(user): Einen bestehenden Benutzer, Mediathekar oder Administrator löschen.

authorize(user): Nach der Registrierung eines Benutzers beziehungsweise Mediathekars den Account freigeben.

Kommunikationspartner

⟨CL101⟩, ⟨CL102⟩

5.2 Implementierung von Komponente $\langle C20 \rangle$: Ressourcenverwaltung

Die Komponente $\langle C20 \rangle$ Ressourcenverwaltung ist für die Verwaltung der Ressourcen, wie Bücher, E-Books, PDFs, Videos und Hardware zuständig.

5.2.1 Paket-/Klassendiagramm

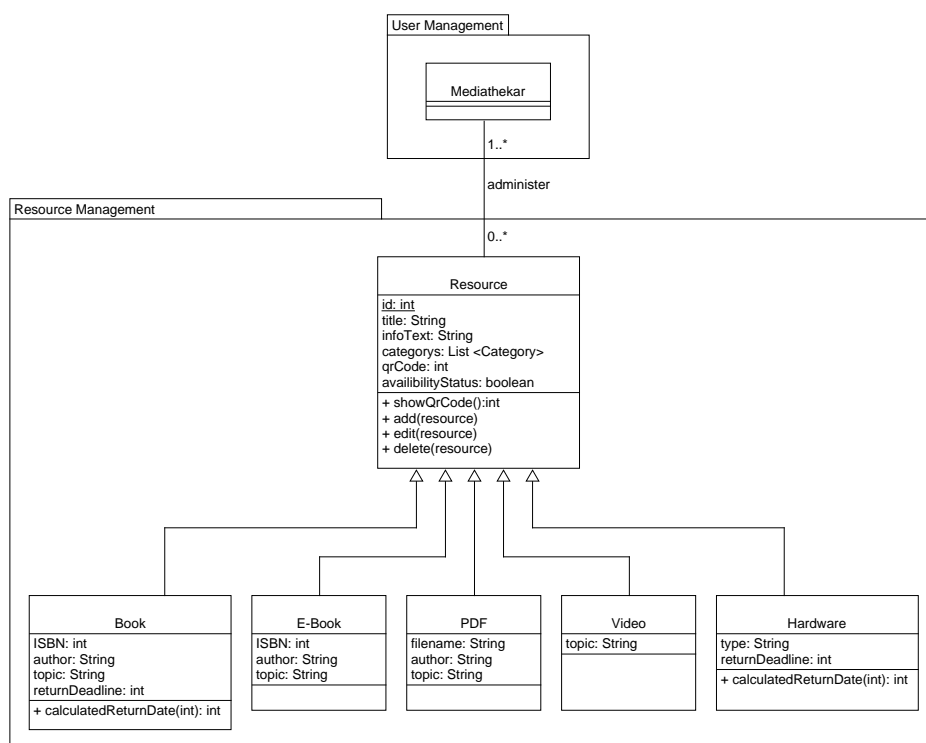


Abbildung 5.2: Klassendiagramm für Komponente $\langle C20 \rangle$

5.2.2 Erläuterung

In der Komponente Ressourcenverwaltung ist die Klasse Ressource, von welcher die Klassen Buch, E-Book, PDF, Video und Hardware erben. Alle Ressourcen werden von einem oder mehreren Mediathekaren bzw. Administrator verwaltet. Sie werden von Mediathekaren bzw. Administrator hinzugefügt, bearbeitet oder auch gelöscht.

Ressource<CL201>

Aufgabe

Hinzufügen, bearbeiten und löschen von verschiedenen Ressourcen.

Attribute

id: Nummer zur eindeutigen Identifizierung

title: Der Titel, beziehungsweise der Name des jeweiligen Ressource. **infoText:** Informationstext zu der Ressource, wie eine Zusammenfassung eines Buches.

categorys: Auflistung der Kategorien der Ressource. **qrCode:** Nummer zur Generierung eines Qr-Codes.

availabilityStatus: Zeigt, ob die Ressource ausleihbar ist oder nicht.

Operationen

showQrCode: Diese Operation dient dazu den QR-Code in eine Zeichenreihenfolge umzuwandeln. **delete:** Mit dieser Operation kann eine Ressource gelöscht werden.

Kommunikationspartner

<CL102>, <CL202>, <CL203>, <CL204>, <CL205>, <CL206>

Buch<CL202>

Aufgabe

Hinzufügen, bearbeiten und löschen von Büchern.

Attribute

ISBN: ISBN des Buches

author: Autor des Buches

topic: Thema des Buches für die Stichwortsuche

returnDeadline: Ausleihfrist

Operationen

calculateReturnDate: Hier wird das Rückgabedatum berechnet.

Kommunikationspartner

<CL201>

E-Book<CL203>

Aufgabe

Hinzufügen, bearbeiten und löschen von E-Books.

Attribute

ISBN: ISBN des E-Books

author: Autor des E-Books

topic: Thema des E-Books für die Stichwortsuche

Operationen

Die Klasse hat keine weiteren Operationen.

Kommunikationspartner

⟨CL201⟩

PDF⟨CL204⟩

Aufgabe

Hinzufügen, bearbeiten und löschen von PDFs.

Attribute

filename: Dateiname der PDF-Datei

author: Autor der PDF **topic:** Thema der PDF für die Stichwortsuche

Operationen

Die Klasse hat keine weiteren Operationen.

Kommunikationspartner

⟨CL201⟩

Video⟨CL205⟩

Aufgabe

Hinzufügen, bearbeiten und löschen von Videos.

Attribute

topic: Thema des Videos für die Stichwortsuche

Operationen

Die Klasse hat keine weiteren Operationen.

Kommunikationspartner

⟨CL201⟩

Hardware $\langle CL206 \rangle$

Aufgabe

Hinzufügen, bearbeiten und löschen von Hardware.

Attribute

type: Art der Hardware **returnDeadline:** Ausleihfrist

Operationen

calculateReturnDate: Hier wird das Rückgabedatum berechnet.

Kommunikationspartner

$\langle CL201 \rangle$

5.3 Implementierung von Komponente $\langle C30 \rangle$: Suchfunktion

Die Komponente (C30) ist für die Steuerung der Suchfunktion der hybriden Mediathek zuständig. Dazu gehört das Auffinden von Ressourcen anhand ihrer eindeutigen ID-Nummern, relevanter Schlüsselwörter oder mithilfe des zugewiesenen QR-Codes.

5.3.1 Paket-/Klassendiagramm

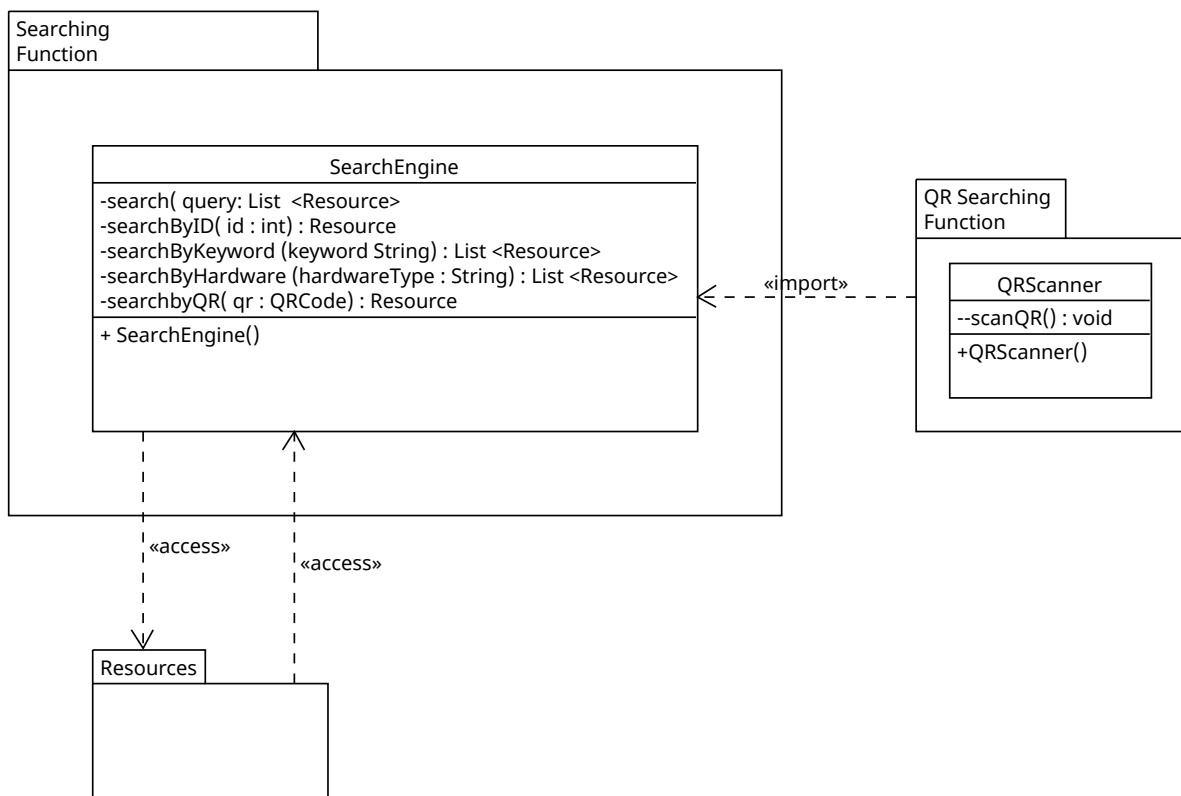


Abbildung 5.3: Klassendiagramm für Komponente $\langle C30 \rangle$

In Abbildung 5.3 wird der Aufbau der Komponenten Suchfunktion nach beliebigen Ressourcen dargestellt. Dieses UML-Klassen- und Paketdiagramm stellt die Komponenten der Suchfunktion innerhalb der Website dar, einschließlich einer QR-Suchoption. Die Klasse **SearchEngine** ist im Paket 'Searching Function' enthalten. Das hilft dabei, die Komponenten der Suchfunktion zu organisieren. Die Klasse **QRCode** ist im Paket "QR Searching Function" enthalten und unterstützt bei der Organisation der Komponenten der QR-Suchfunktion. Die Klasse **QRScanner** wird zum Scannen von QR-Codes mit der Kamera des Geräts des Benutzers verwendet. Die Klasse **Resource** (enthalten in dem Paket **Resource**)

siehe 5.2) verfügt nun über ein qrCode-Attribut, das den mit der Ressource verbundenen QR-Code darstellt.

5.3.2 Erläuterung

SearchEngine⟨CL301⟩

Aufgabe

Suchen und Finden der richtigen Ressource abhängig von der Eingabe.

Attribute

Die Klasse SearchEngine beinhaltet keine Attribute.

Operationen

search(): Nimmt eine Suchanfrage und gibt eine Liste von Ressourcen zurück, die der Anfrage entsprechen

searchByID(): Nimmt eine ID-Nummer als Eingabe und gibt die Ressource mit dieser ID-Nummer zurück

searchByKeyword(): Nimmt ein Schlüsselwort als Eingabe und gibt die Ressourcen zurück, die dieses Schlüsselwort in ihrem Titel oder Autor enthalten

searchByHardware(): Nimmt einen Hardware-Typ als Eingabe und gibt die Ressourcen zurück, die diesen Hardware-Typ für den Zugriff benötigen

searchByQR(): Nimmt einen QR-Code als Eingabe und gibt die mit diesem QR-Code verbundene Ressource zurück

Kommunikationspartner

⟨CL201⟩,⟨CL202⟩,⟨CL203⟩,⟨CL204⟩,⟨CL205⟩,⟨CL206⟩,⟨CL302⟩ QR Scanner

QR Scanner⟨CL302⟩

Aufgabe

Die QR-Codes sollten von der Kamera des Benutzers/der Benutzer gescannt werden können.

Attribute

Die Klasse QR Scanner beinhaltet keine Attribute.

Operationen

scanQR(): void: Diese Methode scannt einen QR-Code mit der Kamera des Geräts des Benutzers.

Kommunikationspartner

⟨CL201⟩,⟨CL202⟩,⟨CL203⟩,⟨CL204⟩,⟨CL205⟩,⟨CL206⟩,⟨CL301⟩

5.4 Implementierung von Komponente $\langle C40 \rangle$: Ausleihverwaltung

Die BorrowManagement-Komponente wird zur Verwaltung des Ausleihprozesses verwendet. Sie ist verantwortlich für die Aufzeichnung verschiedener Informationen im Ausleihprozess wie Ausleiher, ausgeliehene Ressourcen und Ausleihdatum.

5.4.1 Paket-/Klassendiagramm

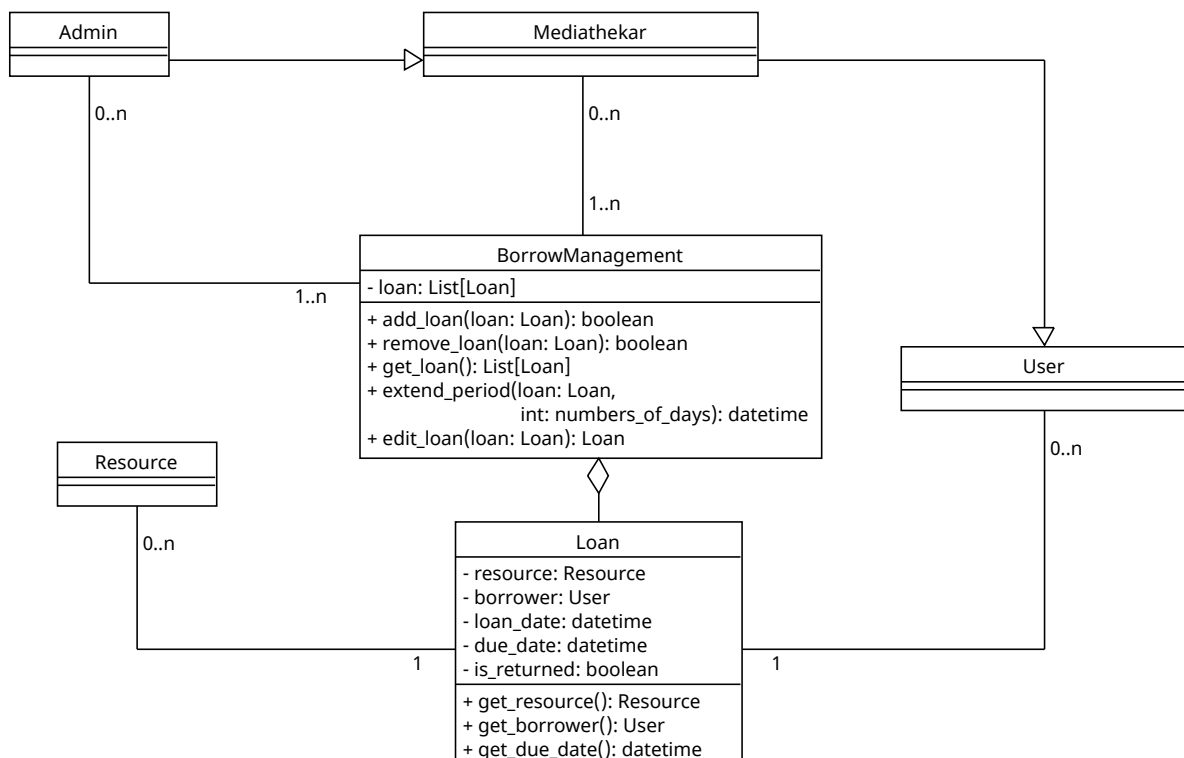


Abbildung 5.4: Klassendiagramm für Komponente $\langle C40 \rangle$

In Abbildung 5.4 wird die Struktur dieser Komponente dargestellt. Die Klasse “Loan” wird verwendet, um Informationen über den Ausleiher und die ausgeliehene Ressource zu speichern. Die Klassen “Admin” und “Mediathekar” verwalten die “Loan”-Objekte mithilfe der Klasse “BorrowManagement”. Dies umfasst Funktionen, wie zum Beispiel die Verlängerung der Ausleihdauer und das Löschen von Ausleihdatensätzen.

5.4.2 Erläuterung

Hier werden die Aufgaben, Attribute, Operationen und Kommunikationspartner der Klassen $\langle CL40 \rangle$ **BorrowManagement** und $\langle CL41 \rangle$ **Loan** erläutert.

BorrowManagement⟨CL40⟩

Aufgabe

Bündelung der Verwaltung des Ausleihprozesses in einem Objekt

Attribute

loan: Liste, die alle Ausleihvorgänge enthält

Operationen

add_loan(loan): Eine Ausleihungsaufzeichnung hinzufügen.

remove_loan(loan): Eine Ausleihungsaufzeichnung löschen.

get_loan(): Die gespeicherte Ausleihliste abrufen.

extend_period(loan, numbers_of_days): Die Frist einer Ausleihung um *numbers_of_days* Tage verlängern.

edit_loan(loan): Eine Ausleihungsaufzeichnung bearbeiten.

Kommunikationspartner

⟨CL103⟩ Admin, ⟨CL102⟩ Mediathekar, ⟨CL41⟩ Loan

Loan⟨CL41⟩

Aufgabe

Die Informationen einer Ausleihung protokollieren

Attribute

resource: Ausgeliehene Ressource

borrower: Ausleiher

loan_date: Ausleihdatum

due_date: Rückgabefrist

is_returned: Status der Rückgabe

Operationen

get_resource(): Ressource, die ausgeliehen wird, abrufen

get_borrower(): Der Ausleiher abrufen

get_due_date(): Rückgabefrist abrufen

Kommunikationspartner

⟨CL201⟩ Resource, ⟨CL101⟩ User, ⟨CL40⟩ BorrowManagement

5.5 Implementierung von Komponente $\langle C50 \rangle$: Favoritenverwaltung

Die Favoritenverwaltung-Komponente wird zur Verwaltung der Favoritenliste verwendet. Sie ist verantwortlich für das Hinzufügen, Entfernen und Abrufen von verschiedenen Ressourcen in der Liste. Benutzer können einigen Ressourcen zur Favoritenliste hinzufügen, löschen usw.

5.5.1 Paket-/Klassendiagramm

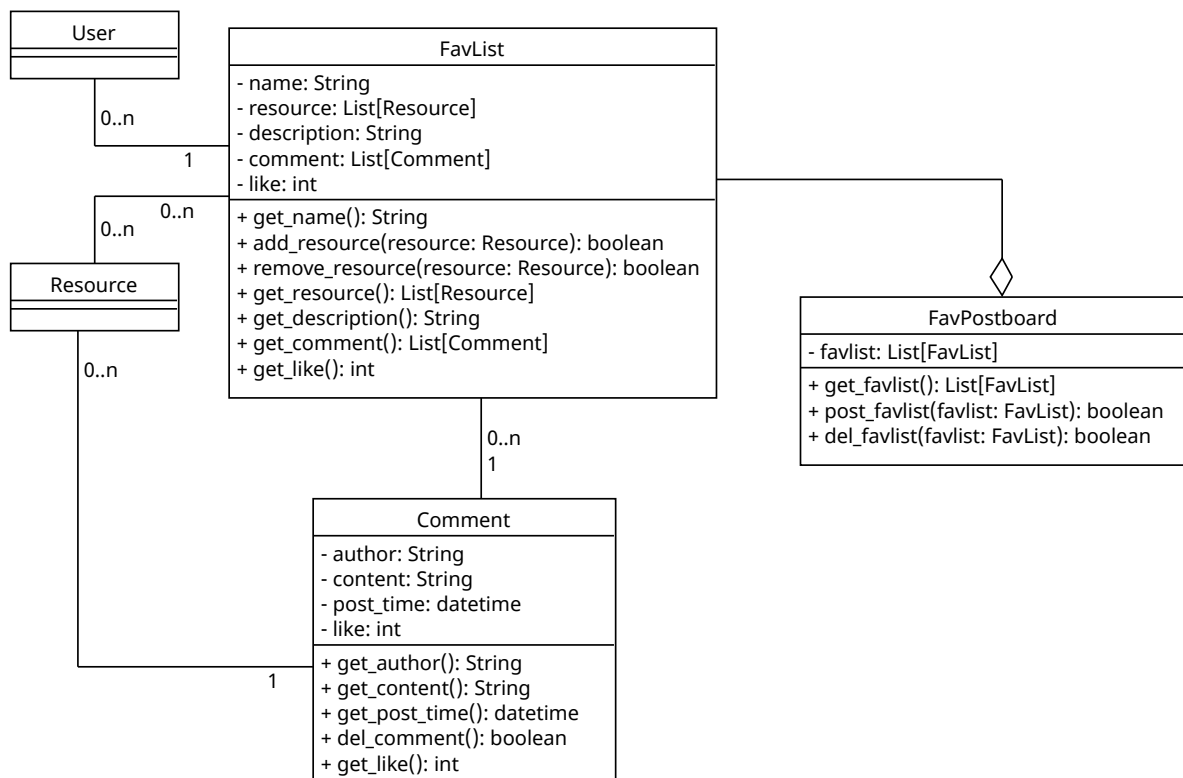


Abbildung 5.5: Klassendiagramm für Komponente $\langle C50 \rangle$

In Abbildung 5.5 wird die Struktur der Favoritenverwaltung-Komponente dargestellt. Die Klasse "Favlist" wird verwendet, um Informationen über die Favoritenliste und die darin enthaltenen Ressourcen zu speichern. Die Klasse "FavPostboard" verwaltet die geteilte Fav-Liste. Dadurch werden Funktionen wie das Hinzufügen, das Entfernen und das Abrufen der Favoritenliste ermöglicht. Die Abbildung veranschaulicht die Beziehungen und Interaktionen zwischen den Komponenten und Klassen innerhalb der Favoritenverwaltung.

5.5.2 Erläuterung

Hier werden die Aufgaben, Attribute, Operationen und Kommunikationspartner der Klassen $\langle CL50 \rangle$ **FavList** und $\langle CL51 \rangle$ **FavPostboard** erläutert.

Favlist $\langle CL50 \rangle$

Aufgabe

Bündelung der eigenen Verwaltung der Fav-Liste in einem Objekt

Attribute

name: Name der Fav-Liste

resource: Liste der hinzugefügten Ressourcen

description: Deskription der Fav-Liste

comment: Liste der Kommentare

like: Anzahl der erhaltenen Likes

Operationen

get_name(): Name der Fav-Liste abrufen.

add_resource(resource): Eine Ressource in diese Fav-Liste hinzufügen.

remove_resource(resource): Eine Ressource von dieser Fav-Liste löschen.

get_resource(): Liste der hinzugefügten Ressourcen abrufen.

get_description(): Deskription der Fav-Liste abrufen.

get_comment(): Liste der Kommentare abrufen.

get_like(): Anzahl der erhaltenen Likes abrufen.

Kommunikationspartner

$\langle CL101 \rangle$ User, $\langle CL201 \rangle$ Resource, $\langle CL51 \rangle$ FavPostboard

FavPostboard $\langle CL51 \rangle$

Aufgabe

Sammlung der geteilten Fav-Listen

Attribute

favlist: Liste der geteilten Fav-Listen

Operationen

get_favlist(): Liste der geteilten Fav-Listen abrufen.

post_favlist(favlist): Eine Fav-Liste teilen.

del_favlist(favlist): Eine Fav-Liste von der Teilen-Liste wegfallen.

Kommunikationspartner

$\langle CL101 \rangle$ User, $\langle CL201 \rangle$ Resource, $\langle CL50 \rangle$ FavList

5.6 Implementierung von Komponente $\langle C60 \rangle$: Benutzeraktivitätenverfolgung

Die Benutzeraktivitätenverfolgungs-komponente verfolgt die Aktivitäten von Benutzern, Administratoren und Mediathekaren z.B. deren letzte Ausleihen von Ressourcen oder Beiträge wie hinzugefügte oder gelöschte Ressourcen.

5.6.1 Paket-/Klassendiagramm

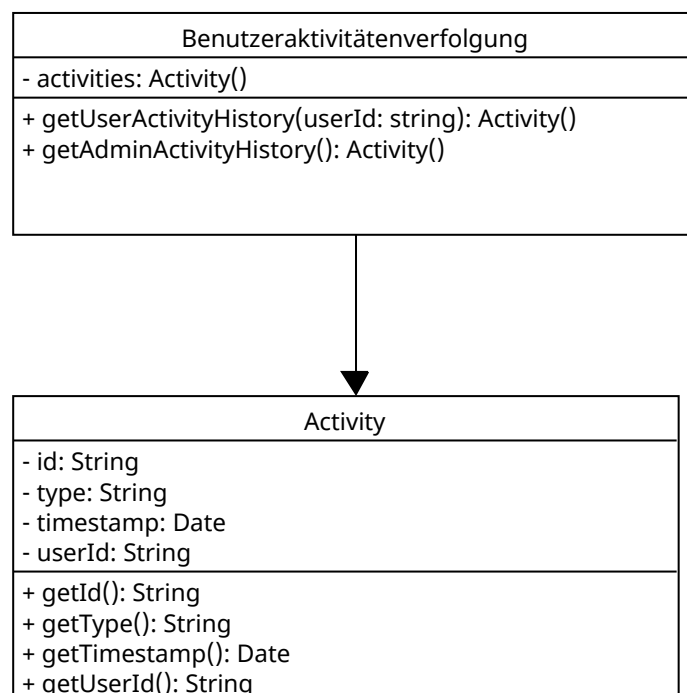


Abbildung 5.6: Klassendiagramm für Komponente $\langle C60 \rangle$

5.6.2 Erläuterung

Hier werden die Aufgaben, Attribute, und Operationen der Klassen $\langle CL60 \rangle$

Benutzeraktivitätenverfolgung und $\langle CL61 \rangle$ **Activity** erläutert.

Benutzeraktivitätenverfolgung $\langle CL60 \rangle$

Aufgabe

Benutzer-, Administator-, und Mediathekaraktivitäten zu verfolgen

Attribute

activities: Liste der letzten Aktivitäten von Benutzern, Administratoren, und Mediathekaren

Operationen

getUserActivityHistory(userId): Die letzten Aktivitäten von Benutzern (z.B. letzte Ausleihe von Ressourcen) abrufen

getAdminActivityHistory(): Die letzten Aktivitäten von Administratoren und Mediathekaren (z.B. letzte Addieren/Löschen von Ressourcen) abrufen

Kommunikationspartner

-

Activity $\langle CL61 \rangle$

Aufgabe

Speicherung aller Daten der letzten Aktivitäten

Attribute

id: ID der Aktivität

type: Aktivitätstyp

timestamp: Zeitstempel der Aktivität

userId: ID von Benutzer

Operationen

getId(): Aktivitäts-ID abrufen

getType(): Aktivitätstyp abrufen

getTimestamp(): Zeitstempel der Aktivität abrufen

getUserId(): Benutzer-ID abrufen

Kommunikationspartner

-

5.7 Implementierung von Komponente $\langle C70 \rangle$: Sprachverwaltung

Die Sprachverwaltungskomponente ermöglicht es allen Benutzertypen, von normalen Benutzern, Mediathekaren, bis zu Administratoren, die Anzeigesprache nach ihren Wünschen zu ändern.

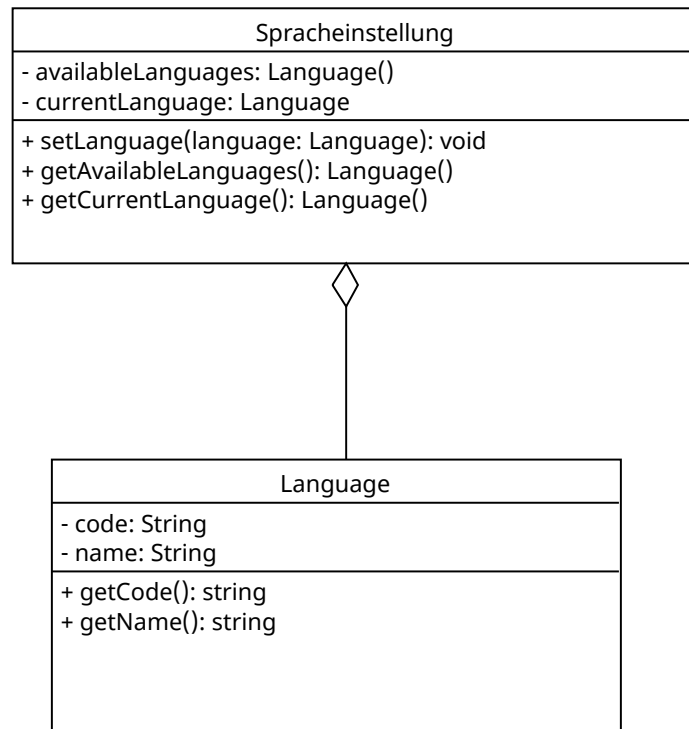


Abbildung 5.7: Klassendiagramm für Komponente <C70>

5.7.1 Paket-/Klassendiagramm

5.7.2 Erläuterung

Hier werden die Aufgaben, Attribute, und Operationen der Klassen <CL70> **Spracheinstellung** und <CL71> **Language** erläutert.

Spracheinstellung<CL70>

Aufgabe

Anzeigesprache nach Wunsch der Benutzern zu ändern.

Attribute

availableLanguages: Liste der verfügbaren Sprachen

currentLanguage: Anzeige der aktuell verwendeten Sprache

Operationen

setLanguage(language): Sprache ändern

getAvailableLanguages(): Liste von verfügbaren Sprachen abrufen

getCurrentLanguage(): Aktuell verwendete Sprache abrufen

Kommunikationspartner

-

Language<CL71>

Aufgabe

Speicherung von verfügbaren Sprachen

Attribute

code: Code(ID) der verfügbaren Sprachen

name: Die Namen der verfügbaren Sprachen

Operationen

getCode(): Der Sprachcode abrufen

getName(): Name von der Sprache abrufen

Kommunikationspartner

-

6 Datenmodell

In diesem Kapitel wird beschrieben, wie das Datenmodell von „Mediaverse“ funktioniert. Die webbasierte Anwendung speichert dauerhaft registrierte Nutzer und Ressourcen in der Django Datenbank. Neben den normalen Benutzern gehören auch die Mediathekare sowie Administratoren dazu. Somit gibt es die vier Entitäten Benutzer, Mediathekar, Administrator und Ressource, welche das Datenmodell von Mediaverse beschreiben. Die Entitäten und ihre Beziehungen zueinander beschreiben den Kern unserer Software und werden in Abbildung 3.1 anhand eines UML-Klassendiagramms dargestellt.

6.1 Diagramm

Die Ressourcen <E10> von Mediaverse besitzen jeweils einen Titel, eine zugehörige Kategorie, einen Author, ein Erstellungsdatum, eine URL, welche als String gespeichert werden. Zudem gibt es eine Liste an Wörter, die als Referenzwörter fungieren und als String gespeichert werden, um nach einer beliebigen Ressource suchen zu können. Des Weiteren besitzen Ressourcen, je nach Art der Ressource einen Barcode und oder QR-Code, welche auch als String gespeichert werden. Ressourcen können je nach Art auch ausleihbar sein bzw. heruntergeladen werden. Der Rückgabewert ist bei beiden boolean und gibt jeweils zurück, ob es möglich ist die Ressource auszuleihen bzw. herunterzuladen oder nicht. Dabei können Benutzer, Mediathekare, und Administratoren beliebig viele Ressourcen ausleihen bzw. herunterladen. Mediathekare und Administratoren können auch beliebig viele Ressourcen anlegen, hinzufügen sowie löschen.

Die Benutzer <E20> von Mediaverse besitzen jeweils einen Benutzernamen, Vornamen, Nachnamen und eine E-Mail sowie ein Passwort, die alle als String gespeichert werden. Dabei wird das Passwort im Mediaverse-System selber jedoch nicht als String gespeichert, sondern verschlüsselt. Jedem Benutzer wird zudem eine ID-Nummer zugewiesen, die als Integer gespeichert wird. Zudem kann der Verlauf der eigenen Aktionen eingesehen werden und eine Favoritenliste erstellt werden, die als Liste von den jeweiligen Ressourcen gespeichert wird. Benutzer können Ressourcen ausleihen, zurückgeben und herunterladen. Dabei wird kein Rückgabewert zurückgegeben. Des Weiteren können beliebig viele Benutzer vom Administrator angelegt, bearbeitet und gelöscht werden.

Mediathekare $\langle E30 \rangle$ in Mediaverse erben die Attribute und Operationen von Benutzern $\langle E20 \rangle$. Zudem können beliebig viele Mediathekare auch Ressourcen anlegen, bearbeiten und löschen. Dazu wird bei jedem Mediathekar eine Liste von Ressourcen gespeichert, die angibt, welche Ressourcen der Mediathekar angelegt hat.

Die Administratoren $\langle E40 \rangle$ von Mediaverse erben die Attribute und Operationen von den Benutzern $\langle E20 \rangle$ und den Mediathekaren $\langle E30 \rangle$. Des Weiteren gilt, dass mindestens ein Administrator die Mediathekare und Benutzer anlegt bzw. verifiziert, sowie bearbeiten und löschen kann.

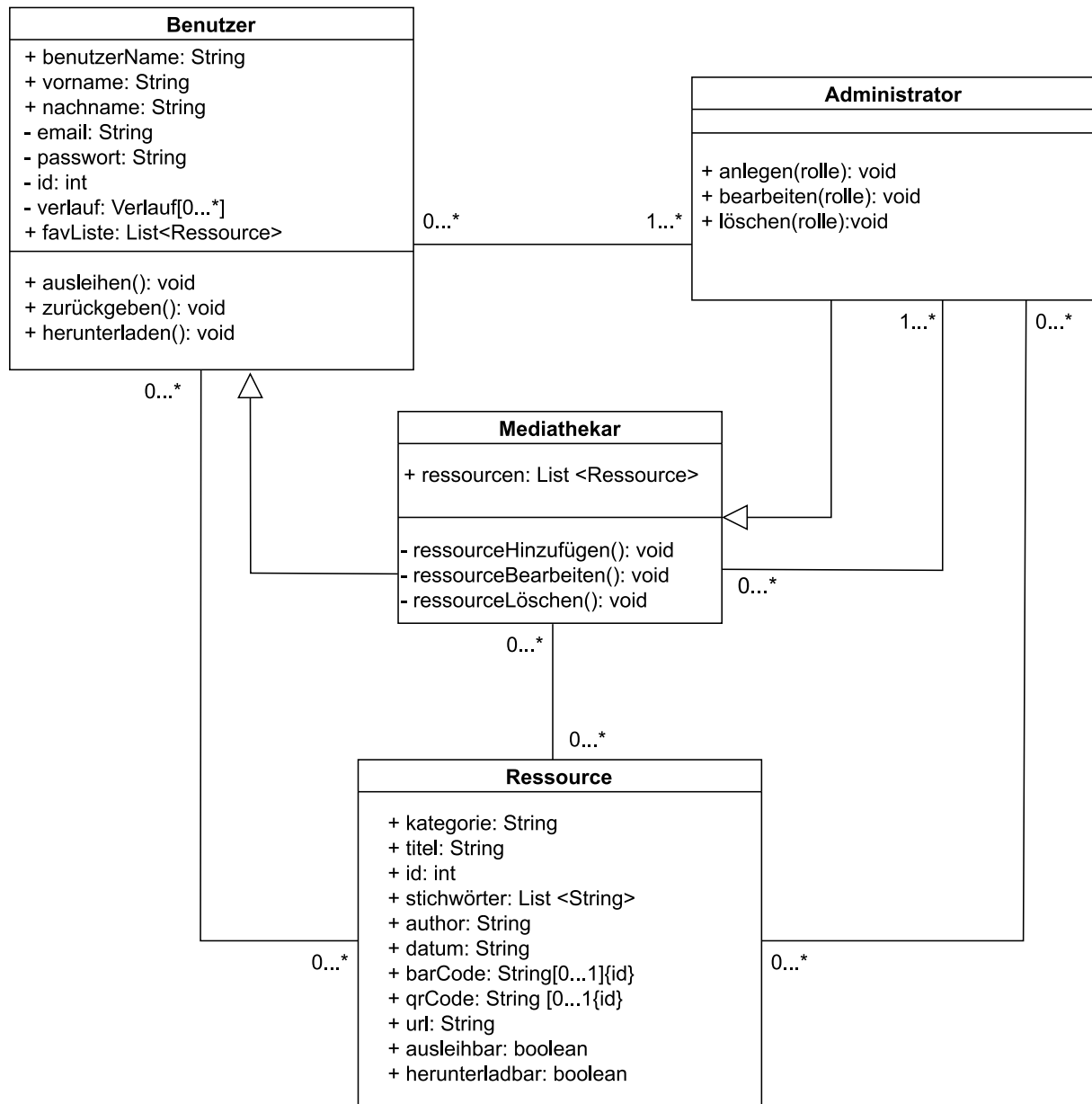


Abbildung 6.1: Klassendiagramm: Projektübersicht

6.2 Erläuterung

Im Folgenden werden die vier Entitäten Benutzer, Mediathekar, Administrator und Ressource in Tabellen genauer erläutert und die Beziehungen zueinander näher beschrieben.

Ressourcen $\langle E10 \rangle$

Ressourcen sind der Kern von Mediaverse, denn sie werden von Benutzern als auch Mediathekaren und Administratoren genutzt bzw. verwaltet.

Beziehung	Kardinalität	Erwartete Datenmenge	Beschreibung
Benutzer $\langle E20 \rangle$	0..*	Mehrere Kilobyte	Einer Ressource können beliebig viele Benutzern zugeordnet werden. Eine Ressource kann von mehreren Benutzern verwendet werden. Zum Beispiel kann sie sich in der Favoritenliste mehrerer Benutzer befinden.
Mediathekar $\langle E30 \rangle$	0..*	Mehrere Kilobyte	Eine Ressource kann beliebig vielen Mediathekaren zugeordnet werden. Eine Ressource kann von mehreren Mediathekaren verwaltet werden. Zum Beispiel können verschiedene Mediathekare die Informationen und den Status bearbeiten.

Benutzer $\langle E20 \rangle$

Neben den Ressourcen bilden die Benutzer die zentrale Entität der webbasierten Anwendung. Schwerpunkt der Beziehungen von Benutzern ist auf die Ressourcen gelegt, dennoch gibt es eine Verwaltungsbeziehung zum Administrator.

Beziehung	Kardinalität	Erwartete Datenmenge	Beschreibung
Ressourcen $\langle E10 \rangle$	0..*	Mehrere Kilobyte	Einem Benutzer können beliebig viele Ressourcen zugewiesen werden. Ein Benutzer kann gleichzeitig mehrere Ressourcen nutzen. Zum Beispiel kann er eine Ressource herunterladen, gleichzeitig eine andere ausleihen und noch eine andere in der Fav-Liste legen.

Mediathekar $\langle E30 \rangle$

Mediathekare bilden die Verwaltungsentität der Ressourcen, aber werden selber vom Administrator verwaltet.

Beziehung	Kardinalität	Erwartete Datenmenge	Beschreibung
Ressourcen $\langle E10 \rangle$	0..*	Mehrere Kilobyte	Einem Mediathekar können beliebig viele Ressourcen zugeordnet werden. Ein Mediathekar verwaltet gleichzeitig mehrere Ressourcen. Er kann eine Ressource hinzufügen, eine andere löschen und die Informationen anderer Ressourcen bearbeiten.

6.2.1 Verwaltung der Konten

Administrator $\langle E40 \rangle$

Administratoren verwalten den Benutzer, den Administrator und die Ressource.

Beziehung	Kardinalität	Erwartete Datenmenge	Beschreibung
Benutzer $\langle E20 \rangle$	0..*	Mehrere Kilobyte	Einem Administrator können beliebig viele Benutzer zugeordnet werden. Ein Administrator kann gleichzeitig mehrere Benutzer verwalten. Er kann mehrere Benutzerkonten hinzufügen, einige löschen und deren Informationen bearbeiten.
Mediathekar $\langle E30 \rangle$	0..*	Mehrere Kilobyte	Einem Administrator können beliebig viele Administratoren zugewiesen werden. Ein Administrator kann gleichzeitig mehrere Mediathekaren verwalten. Er kann mehrere Mediathekarkonten hinzufügen, löschen und deren Informationen bearbeiten.

7 Konfiguration

Konfigurationsdateien spielen eine entscheidende Rolle bei der Anpassung des Verhaltens der Website. Unser Mediaverse verwendet Django für das Backend, Svelte für das Frontend, die auf einem Linux-basierten Server laufen und PostgreSQL als Datenbankmanagementsystem in unserer Serverinfrastruktur. Es fungiert als Datenspeicher- und Verwaltungsschicht zwischen dem Server und dem Backend unserer Website. Der Server kommuniziert mit PostgreSQL, um Daten effektiv zu speichern, abzurufen und zu verwalten. Dieses System gewährleistet eine robuste und anpassbare Plattform, die Sicherheit, Zuverlässigkeit und Skalierbarkeit bietet.

Zum Zeitpunkt der Erstellung dieses Dokuments werden die notwendigen Komponenten für einen reibungslosen Betrieb unseres Projekts auf elektronischen Geräten wie Tablets oder Laptops ermittelt. Da sich das Team in der Entscheidungs- und Umsetzungsphase befindet, finden sich hier einige Optionen für optimale Leistung und Systemanforderungen, an denen das Team gearbeitet hat, um einen Einblick in die Arbeit zu geben:

1. Webserver-Konfiguration:

- Konfigurationsdatei: *webserver.conf*
- Vorgesehener Speicherort: */path/to/project/config/webserver.conf*
- Zweck: In dieser Konfigurationsdatei werden die Servereinstellungen festgelegt, z. B. die maximale Anzahl der gleichzeitigen Verbindungen, Caching-Optionen und Komprimierungseinstellungen. Damit wird sichergestellt, dass der Webserver, auf dem die Website gehostet wird, richtig konfiguriert ist, um eingehende Anfragen zu bearbeiten und Webseiten effizient zu liefern.

2. Browser-Kompatibilitätseinstellungen:

- Konfigurationsdatei: *'browserCompatibility.conf'*
- Vorgesehener Speicherort: *'/path/to/project/config/browserCompatibility.conf'*
- Zweck: Diese Konfigurationsdatei enthält Informationen über empfohlene Browserversionen und Kompatibilitätsrichtlinien. Sie kann Einzelheiten über unterstützte Browsertechnologien, CSS-Präfixe und JavaScript-Funktionen enthalten. Diese Einträge tragen dazu bei, dass die Website in verschiedenen Browsern und Versionen optimal funktioniert.

3. Server-seitige Skripting-Konfiguration:

- Konfigurationsdatei: `'serverScripting.conf'`
- Vorgesehener Speicherort: `/path/to/project/config/serverScripting.conf`.
- Zweck: Diese Konfigurationsdatei definiert Einstellungen für serverseitige Skriptsprachen wie PHP oder Python. Sie kann Optionen wie Fehlerbehandlung, Ressourcenbegrenzung und Sicherheitseinstellungen enthalten. Diese Einträge stellen sicher, dass die serverseitigen Skripte, die die Website betreiben, reibungslos und sicher funktionieren.

Darüber hinaus werden Parameter entworfen und angepasst, um verschiedene Aspekte der Website zu konfigurieren. Hierzu zählen die Anzahl der aufgelisteten Elemente pro Seite in den Suchergebnissen, die Standarddauer für das Ausleihen von Ressourcen und die maximale Dateigröße, die von den Nutzern hochgeladen werden kann, beispielsweise beim Hinzufügen neuer Ressourcen oder Profilbilder.

In den kommenden Wochen werden weitere Fortschritte erzielt, und auf dieses Dokument wird verwiesen, um detailliertere Informationen bereitzustellen.

8 Änderungen gegenüber Fachentwurf

In diesem Kapitel werden die Änderungen erklärt, die sich mit der Umsetzung des Projektes gegenüber zum Fachentwurf erschlossen haben.

- Anpassung der Funktionen: Die Kriterien $\langle RM20 \rangle$ und $\langle RC2 \rangle$ werden im Rahmen dieses Projektes nicht umgesetzt.
- Die Benutzerkonten müssen vor der Nutzung aktiviert werden, anstatt sofort aktiv zu sein. Dies geschieht durch den Administrator, der die Konten aktiviert.
- Anmerkung zur Funktionalität F60: Hinzufügen, Bearbeiten, Löschen, der für diese Rolle zuständige Akteur wurde zum Mediathekar korrigiert (auch in vorherigen Dokumenten).

9 Erfüllung der Kriterien

In diesem Kapitel werden die einzelnen Kriterien aus dem Pflichtenheft erneut betrachtet. Im Einzelnen werden alle Kriterien (Soll-, Muss- und Kannkriterien) den entsprechenden Komponenten zugeordnet, die zur Realisierung benötigt werden.

9.1 Musskriterien

Die folgenden Kriterien sind unabdingbar und müssen durch das Produkt erfüllt werden:

$\langle RM1 \rangle$:

Mediaverse muss drei verschiedene Nutzertypen unterstützen: Benutzer, Mediathekare und Administrator. Dies wird mithilfe der $\langle C10 \rangle$ Benutzerverwaltungskomponente umgesetzt.

$\langle RM2 \rangle$:

Mediaverse muss eine Sign in-Funktion für Benutzer bereitstellen. Die $\langle C10 \rangle$ Benutzerverwaltungskomponente ist hierfür zuständig.

$\langle RM3 \rangle$:

Mediaverse muss eine Log in-Funktion für Benutzer bereitstellen. Die Funktion wird mithilfe der $\langle C10 \rangle$ Benutzerverwaltungskomponente realisiert.

$\langle RM4 \rangle$:

Benutzer müssen in der Lage sein, die Verfügbarkeit von digitalen und physischen Ressourcen zu prüfen. Hierfür ist die $\langle C40 \rangle$ Ausleihverwaltungskomponente verantwortlich.

$\langle RM5 \rangle$:

Benutzer müssen in der Lage sein, verfügbare Ressourcen auszuleihen. Diese Funktion wird mit der $\langle C40 \rangle$ Ausleihverwaltungskomponente umgesetzt.

$\langle RM6 \rangle$:

Mediaverse muss es Benutzern ermöglichen, Ressourcen mithilfe von QR-Codes zu suchen und auszuleihen. Die Umsetzung wird durch die $\langle C30 \rangle$ Suchfunktionskomponente gewährleistet.

$\langle RM7 \rangle$:

Mediaverse muss es Benutzern ermöglichen, Ressourcen mithilfe von Identifikationsnummern zu suchen und auszuleihen. Die $\langle C30 \rangle$ Suchfunktionskomponente sowie die $\langle C40 \rangle$ Ausleihverwaltungskomponente setzen diese Funktionen um.

$\langle RM8 \rangle$:

Mediaverse muss es Benutzern ermöglichen, Favoritenlisten zu verwalten. Hier ist die $\langle C50 \rangle$ Favoritenverwaltungskomponente verantwortlich.

$\langle RM9 \rangle$:

Mediaverse muss die Suche nach Ressourcen durch Filtern erleichtern. Die Komponente $\langle C30 \rangle$ Suchfunktion stellt diese Funktion sicher.

$\langle RM10 \rangle$:

Mediaverse muss digitale Kopien von Ressourcen zum Download anbieten. Mithilfe der $\langle C20 \rangle$ Ressourcenverwaltung wird dies realisiert.

$\langle RM11 \rangle$:

Mediaverse muss es Benutzern ermöglichen, verschiedene Ressourcentypen zu nutzen. Auch hier wird die Funktion mit der $\langle C20 \rangle$ Ressourcenverwaltung umgesetzt.

$\langle RM12 \rangle$:

Mediaverse muss eine Log in-Funktion für Mediathekare bereitstellen. Hierfür ist die $\langle C10 \rangle$ Benutzerverwaltungskomponente zuständig.

$\langle RM13 \rangle$:

Mediathekare müssen in der Lage sein, Ressourcen anzulegen. Mit der $\langle C20 \rangle$ Ressourcenverwaltung wird dies gewährleistet.

$\langle RM14 \rangle$:

Mediathekare müssen in der Lage sein, Ressourcen zu editieren. Die $\langle C20 \rangle$ Ressourcenverwaltungskomponente ist auch hierfür verantwortlich.

$\langle RM15 \rangle$:

Mediathekare müssen in der Lage sein, Ressourcen zu löschen. Diese Funktion wird auch mit der $\langle C20 \rangle$ Ressourcenverwaltungskomponente erfüllt.

$\langle RM16 \rangle$:

Mediaverse muss eine Log in-Funktion für Administrator bereitstellen. Dies wird mithilfe der $\langle C10 \rangle$ Benutzerverwaltungskomponente umgesetzt.

$\langle RM17 \rangle$:

Administrator muss in der Lage sein, Benutzer anzulegen, editieren und löschen. Auch diese Funktionen werden mit der $\langle C10 \rangle$ Benutzerverwaltungskomponente realisiert.

⟨RM18⟩ :

Administrator muss in der Lage sein, Mediathekare anzulegen, editieren und löschen. Hierfür ist auch die ⟨C10⟩ Benutzerverwaltungskomponente zuständig.

⟨RM19⟩ :

Mediaverse muss es dem Administrator ermöglichen, Berichte über alle Aktionen zu generieren. Die ⟨C60⟩ Benutzeraktivitätenverfolgung stellt diese Funktionalität sicher.

⟨RM20⟩ :

Mediaverse muss es dem Administrator ermöglichen, neue Arten von Ressourcentypen einfach hinzuzufügen. Diese Funktion wird nicht umgesetzt.

⟨RM21⟩ :

Mediaverse muss es dem Administrator ermöglichen, bestehende Ressourcentypen zu ändern. Auch diese Funktion wird durch die Komponente ⟨C20⟩ "Ressourcenverwaltung" gewährleistet.

⟨RM22⟩ :

Mediaverse muss es dem Administrator ermöglichen, bestehende Ressourcentypen zu entfernen. Hier ist auch die ⟨C20⟩ Ressourcenverwaltungskomponente verantwortlich.

9.2 Sollkriterien

Die Erfüllung folgender Kriterien für das abzugebende Produkt wird angestrebt:

⟨RS1⟩ :

Mediaverse soll eine erweiterte Suche nach Ressourcen anbieten, einschließlich Filtern nach Typ, Verfügbarkeit und anderen Merkmalen. Diese Funktionalitäten sollen mit der ⟨C30⟩ Suchfunktionskomponente umgesetzt werden.

⟨RS2⟩ :

Mediaverse soll eine Funktion zur Verlängerung von Ausleihen bereitstellen. Mithilfe der ⟨C40⟩ Ausleihverwaltung soll diese Funktion gewährleistet sein.

⟨RS3⟩ :

Die Website soll in deutscher und englischer Sprache angeboten werden. Dies wird mit der ⟨C70⟩ Sprachverwaltungskomponente realisiert.

9.3 Kannkriterien

Die Erfüllung folgender Kriterien für das abzugebende Produkt wird angestrebt:

⟨RC1⟩ :

Es kann eine Empfehlungsfunktion für ähnliche Ressourcen hinzugefügt werden. Die ⟨C30⟩ Suchfunktionskomponente ist hierfür verantwortlich.

⟨RC2⟩ :

Es kann eine Möglichkeit zur Bewertung und Kommentierung von Ressourcen durch Benutzer hinzugefügt werden. Diese Funktion wird nicht umgesetzt.

⟨RC3⟩ :

Es kann eine Option für die Reservierung von Ressourcen hinzugefügt werden. Mithilfe der ⟨C40⟩ Ausleihverwaltungskomponente wird dies sichergestellt.

⟨RC4⟩ :

Die Webseite kann in verschiedenen Sprachen angeboten werden, um eine breitere Zielgruppe anzusprechen. Dazu zählen zum Beispiel Türkisch, Arabisch und Chinesisch. Hier kann die Komponente ⟨C70⟩ Sprachverwaltung diese Funktionalität gewährleisten.

⟨RC5⟩ :

Mediaverse ermöglicht die Anzeige von Nutzungsstatistiken, die Aufschluss darüber geben, wie oft eine Ressource aufgerufen oder heruntergeladen wurde. Dies wird durch die ⟨C20⟩ Ressourcenverwaltung gesichert.

10 Glossar

Django: Ein Backend-Webframework, das in der Programmiersprache Python geschrieben ist und die schnelle Entwicklung von Webanwendungen ermöglicht.

Svelte: Ein Frontend-Webframework, das es ermöglicht, Webanwendungen mit hoher Leistung und geringem Ressourcenbedarf zu erstellen.

PostgreSQL: ein Open-Source relationales Datenbankmanagementsystem, das für seine Robustheit und Skalierbarkeit bekannt ist. Es bietet fortgeschrittene Funktionen und unterstützt komplexe SQL-Abfragen.

RESTful APIs: (Representational State Transfer) sind eine Reihe von Prinzipien und Konventionen, die bei der Entwicklung von Webanwendungen verwendet werden, um die Kommunikation zwischen Client und Server zu ermöglichen. Sie ermöglichen es, Ressourcen auf standardisierte Weise anzufordern, zu manipulieren und zu übertragen, indem HTTP-Methoden wie GET, POST, PUT und DELETE verwendet werden.

WSGI: WSGI ist der Hauptstandard in Python zur Kommunikation zwischen Webservern und Anwendungen, unterstützt jedoch nur synchronen Code.

DB-connector: Ein DB Connector API ist eine Schnittstelle, die es dem Betriebssystem ermöglicht, auf die Datenbank zuzugreifen und Operationen wie Datenabruf und -speicherung durchzuführen.