

RateYourStuff



Projekt, um verschiedene Medien zu Bewerten, darüber zu diskutieren und Fortschritte dazu anzulegen

Das Projektteam:

- Robin Beck
- Christoph Frischmuth
- Mickey Knop
- John Klippstein

Nutzerdaten zur Anmeldung im Frontend

Nutzername	Passwort	Rolle	enabled
mikmin	admin	Admin	true
avartos	admin	Admin	true
m_rhöse	user	User	true
Rolf_Kruse	user	User	false

Installation

Backend

Datenbank

Der Server ist konfiguriert, um mit einer mariaDB Datenbank zu kommunizieren. Hierbei sind folgende Eckpunkte wichtig:

- Datenbankname: ratedatabase (wird automatisch angelegt)
- Serveradresse: localhost
- Serverport: 3603
- Nutzername: root
- Passwort: (kein Passwort)

Hinweis: Für das Durchlaufen der Testfälle, wird eine H2 Datenbank namens "ratedatabasetest" angelegt. Somit ist für das Durchführen der Tests keine aktive MariaDB Datenbank erforderlich.

REST-API

Nachdem ein entsprechender Datenbankserver bereitgestellt wurde, kann die Spring-Anwendung gestartet werden. Beim Start der Anwendung werden die Tabellen, sofern noch nicht vorhanden, automatisch angelegt. Sobald die Tabellen angelegt wurden, kann das Script "[data.sql](#)", welches sich im src-Ordner befindet, über z.

B. phpmyadmin importiert werden. (in dem Skript befinden sich die Testdaten. Die Testdaten bitte nur in die Datenbank "ratedatabase" importieren.

Der Server ist anschließend unter dem Port 5000 erreichbar.

Über [Swagger](#) können die zur Verfügung stehenden Schnittstellen eingesehen werden.

Frontend

Um das Frontend zu nutzen, gibt es zwei Varianten:

Deployte Version

Über [Github-Pages](#) ist eine fertig deployte Version des Frontends verfügbar. Die Nutzung des deployten Frontends hat den Vorteil, dass weder nodejs noch andere zusätzliche Module installiert werden müssen. Wichtig ist lediglich, dass das Backend sowie die Datenbank aktiv sind.

Manuelle Installation

Das Frontend kann über das [Frontend-Repository](#) heruntergeladen werden. Bitte beachten Sie, dass zum Ausführen des Frontends nodejs benötigt wird. Sobald das Frontend heruntergeladen wurde, müssen mithilfe des Befehls

```
npm install
```

fehlende Abhängigkeiten nachinstalliert werden. (Die Abhängigkeiten werden automatisch geladen und installiert, sobald der Befehl eingegeben wurde.) Sobald alle Pakete nachinstalliert wurden, kann das Frontend mithilfe von

```
npm start
```

gestartet werden. Für gewöhnlich sollte sich ein Browserfenster mit der Startseite der Anwendung öffnen. Sollte dies nicht der Fall sein, so ist das Frontend über: <http://localhost:3000> erreichbar.

Da sowohl Front- als auch Backend auf dem selben Server betrieben werden, kann es zu Problemen durch CORS kommen. CORS ist zwar im Backend für diesen Fall deaktiviert worden, kann aber ggf. dennoch Probleme verursachen. In diesem Fall sollte eine entsprechende Browsererweiterung nachinstalliert werden, um CORS Fehler zu vermeiden.

Einleitung

Die Welt dreht sich immer mehr um ... Die Welt dreht sich immer mehr um die verschiedensten Unterhaltungsmedien und es fällt dem Konsumenten immer schwerer sich vorab eine Meinung zu bilden. Vor allem im Jahre 2020, wo das Wort „Fakenews“ nicht nur die Wortneuschöpfung eines sehr verrückten und geistig labilen Politikers ist.

Die meisten veröffentlichten Meinung sind von Kritikern, die evtl. nicht den eigenen Geschmack vertreten und wenn andere Konsumenten ihre Bewertung auf einer Plattform veröffentlichen können, ist diese Plattform meist nur für ein bestimmtes Medium.

Auf unserer Plattform RateYourStuff werden alle Unterhaltungsmedien vertreten, von Büchern bis hin zu Videospielen, können sie zu allem und jedem ihre eigene freie Meinung äußern. (Vorausgesetzt sie gefällt uns und wir löschen sie nicht).

Mit RateYourStuff soll ein persönliches Bewertungssystem für Unterhaltungsmedien verschiedener Art entwickelt werden. Diese werden von den Benutzern des Systems nach verschiedenen Kriterien bewertet und daraus wird ein Finaler Punktwert generiert, der die Qualität bzw. den Unterhaltungswert des Mediums wiedergibt. Darüber hinaus kann der Benutzer seinen Fortschritt zu dem Produkt angeben was beispielsweise die Seite eines Buches oder die Folge einer Serie beinhaltet, anhand dessen wird er in den dazugehörigen Diskussionsforen vor Spoilern automatisch gewarnt. Des Weiteren kann man sich dazugehörige Werke anzeigen und nach Erscheinungsdatum, oder diese auch in einer chronologischen Reihenfolge, filtern.

Ausgangssituation

Im Rahmen der Pandemie, im Jahre 2020 n.Chr. ... Im Rahmen der Pandemie, im Jahre 2020 n.Chr., wurden, gefühlt, unendlich viele Medien konsumiert. Der gewohnte Austausch auf der Arbeit, in der Schule, an der Universität oder in den Kreisen der Freunde viel jedoch aus. Nun hat der Mensch das dringende Bedürfnis sich mitzuteilen und viele Menschen tun ihre Meinung kund, ohne gefragt zu werden.

Nun existiert schon eine größere Menge an Portalen, die sich mit der Bewertung von Medien beschäftigen, aber auch hier zeigen sich Probleme. Andere Portale sind oft nur für ein Medium ausgelegt und, oder sind einfach unübersichtlich, nicht gut organisiert oder bieten nicht die gewünschten Funktionen. An dieser Stelle soll das Konzept von „Rate Your Stuff“ greifen. „RYS“ bietet die Möglichkeit für die populärsten Medien Meinungen abzugeben und Wertungen. Die Meinungen und Wertungen können von anderen Usern/Freunden kommentiert werden und so kann ein Austausch über unterschiedliche Medien, auf einer Plattform stattfinden

Was ist RateYourStuff?

RateYourStuff soll eine Bewertungsplattform für verschiedene Arten von Medien sein. Es ist geplant, folgende Medientypen vorerst zu unterstützen:

- Bücher
- Filme
- Serien
- Videospiele

Benutzer können Medien mit unterschiedlichen Kriterien bewerten. Für jedes Medium gibt es eine allgemeine Bewertung, die durch die Bewertungskriterien bestimmt wird. Benutzer können ihren Fortschritt auf verschiedenen Medientypen teilen. Jedes Medium hat einen eigenen Kommentarbereich. Darüber hinaus ist eine Art Mediensammlung geplant, um Medien logisch zu gruppieren.

Zielbestimmungen

Das System soll eine Plattform zum Bewerten von diversen Mediatypen darstellen. Zu den zu bewertenden Media-Typen zählen Bücher, Filme, Serien – inklusive Episoden – sowie Videospiele. Nutzer sollen ihre Medien-Fortschritte für ihr Profil verwalten und anzeigen lassen können. Für jedes Medium sollen Kommentare geschrieben und diese auch entsprechend kommentiert werden können. Den Nutzern soll es möglich sein untereinander die eigenen Fortschritte zu teilen.

Das System soll in mehreren Etappen entwickelt werden:

- Grundsystem
- Datenbankanbindung
- Webfrontend

Für das Modul Java 02 wird auf dem aus Java 01 entnommenen Grundsystem aufgebaut. Dieses soll so erweitert werden, dass eine Datenbankpersistierung sowie eine Nutzung des Systems über ein Frontend möglich gemacht werden.

Muss-Kriterien

Die nachfolgenden Musskriterien wurden im Vergleich zu Java 01 leicht abgeändert. Entfallene Musskriterien werden mit (Java01) gekennzeichnet. Grund für die Abwandlung waren Zeit und Aufwand des Projekts.

Verwaltung von Medien Das System muss den Nutzern die Möglichkeit bieten: - Medien anzulegen

- Medien zu suchen
- Medien zu bearbeiten
- Medien in Kollektionen einzuteilen

Verwaltung von Fortschritten (Java01) Das System muss den Nutzern die Möglichkeit bieten: - eigene Fortschritte für Medien anzulegen (Java01)

- den aktuellen Fortschritt prozentual oder Medienspezifisch (Buchseite, Episode, Filmminute, Spielzeit) anzuzeigen (Java01)
- bereits angelegte Fortschritte zu ändern (Java01)
- bereits angelegte Fortschritte zu löschen (Java01)

Verwaltung von Kommentaren Das System muss den Nutzern die Möglichkeit bieten: - Kommentare zu einem Medium zu schreiben

- auf die Kommentare von anderen Nutzern zu antworten
- Kommentare mit potenziellen Spoilern auszublenden (Java01)
- Kommentare zu löschen
- Kommentare zu deaktivieren (Java01)
- Kommentare wieder zu öffnen (Java01)

Verwaltung von Kollektionen Das System muss den Nutzern die Möglichkeit bieten: - Kollektionen anzulegen

- Medien zu Kollektionen zuzuweisen
- Medien aus Kollektionen entfernen
- Kollektionen entfernen

Bewerten von Medien Das System muss den Nutzern und Moderatoren die Möglichkeit bieten: - eine Bewertung für ein Medium abzugeben

- die eigene Bewertung zu editieren
- die eigene Bewertung zu löschen
- eine aus den bisher abgegebenen Bewertungen ermittelte Durchschnittsbewertung anzeigen lassen

Verwalten von Benutzern Das System muss den Nutzern und Moderatoren die Möglichkeit bieten: - andere Nutzer anhand ihres Nutzernamens zu suchen (Java01)

- die eigenen Nutzerdaten einzusehen
- die eigenen Nutzerdaten zu ändern
- sich am System zu registrieren
- Nutzer-Accounts zu deaktivieren
- Nutzer-Accounts zu reaktivieren
- das Passwort von Benutzern zurückzusetzen
- Benutzern Moderator-Rechte zu geben

Wunsch-Kriterien

Es ist wünschenswert, das System zu einem späteren Zeitpunkt um folgende Kriterien zu erweitern: - Anzeigen von empfohlenen Medien auf Basis der bereits genutzten / bewerteten Medien

- gezieltes Ausblenden von Medien, die man nicht sehen möchte (z. B.: Anhand des Genres)
- Private Nachrichten an andere Nutzer senden
- Andere Benutzer zu Freundesliste hinzufügen
- Medien abonnieren, um bei Aktualisierungen / Neuerscheinungen benachrichtigt zu werden
- Filtern und Suchen nach Fortschritten, die Nutzer zu Medien angelegt haben
- Eigene Videos, die Ausschnitte des Mediums zeigen, im Rahmen des Zitatrechts, sollen als Kommentar beziehungsweise als Meinung möglich sein. So können, zum Beispiel, Spielabschnitte zitiert und kommentiert werden, um seinen Freunden oder Interessierten eine Möglichkeit zu bieten die Bildung der Meinung nachvollziehen zu können.
- Ein weiteres Wunschkriterium ist eine Merkliste, in der Medien hinterlegt werden können, welche man sich später ansieht/liest/spielt.
- Verwalten von Fortschritten sowie das Teilen von Fortschritten mit anderen Nutzern

Abgrenzungskriterien „Rate Your Stuff“ soll keine herkömmliche Produktbewertung sein. Die Plattform gibt verschiedene Kriterien vor, die sich mit den Inhalten beschäftigen und nicht mit dem Auslieferungszustand oder der Qualität des Papiers.

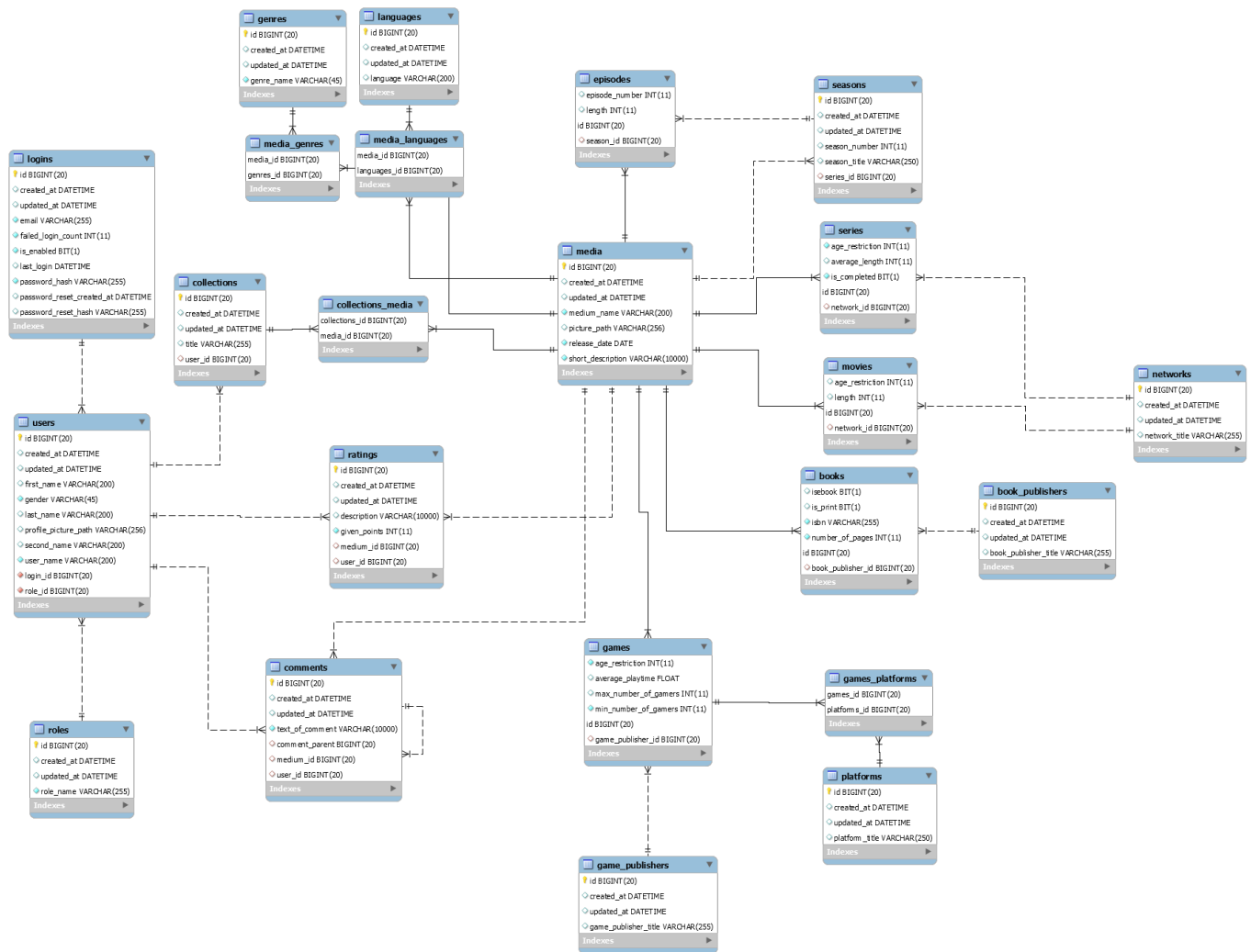
Nicht realisiert werden, sollen:

- Möglichkeit zur Wiedergabe der Medien auf der Seite
- Ermitteln von Bewertungen für Medien auf anderen Seiten
- Trennung von Benutzerbewertungen und Journalistischen Rezensionen

Er - Diagramm

Die Datenbank wurde anhand der in Java definierten Klassen generiert. Nachfolgend befindet sich das resultierende ER-Diagramm:

ER-Diagramm

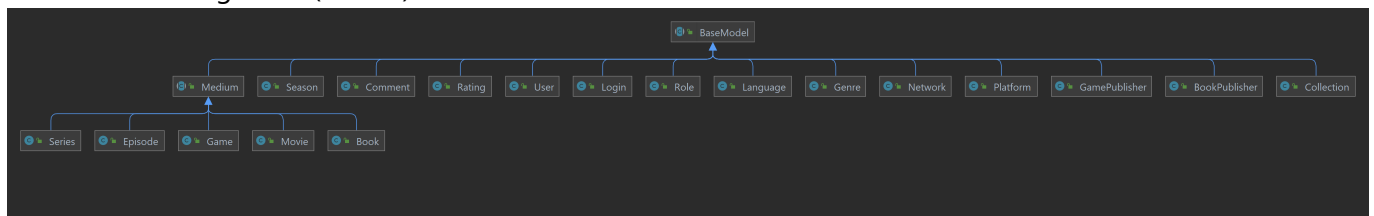


UML-Diagramme

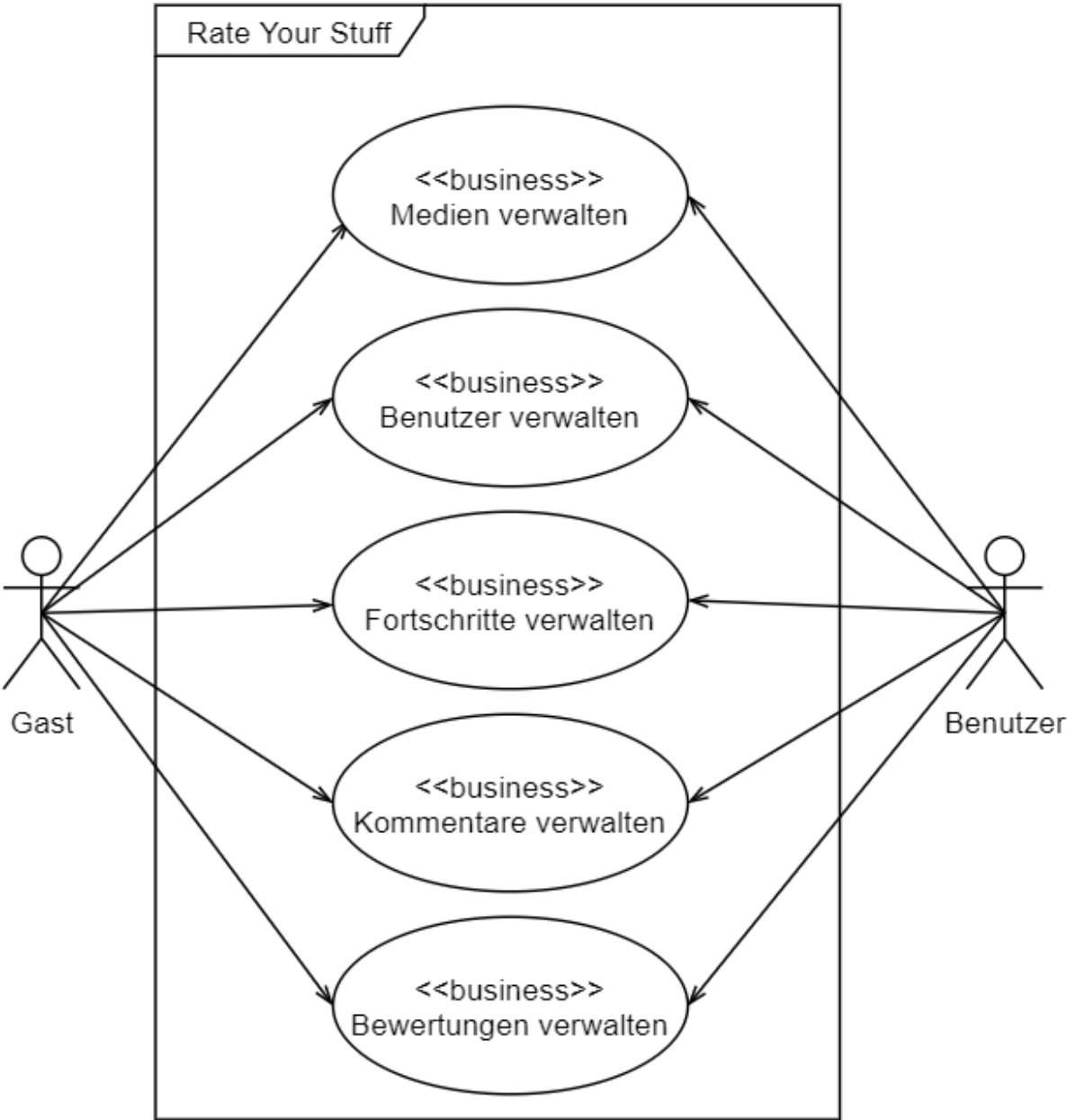
Unter diesem Punkt befinden sich unsere UML-Diagramme die wir im Vorfeld im Modul Java01 erstellt haben um unsere Idee von der Applikation grafisch darzustellen und um eine bessere Vorstellung des Ganzen zu bekommen. Alle UML-Diagramme, welche in Java02 nicht mehr verwendet wurden, haben als Kennzeichnung ein Java01 am Ende.

UML-Klassen Diagramm (Java01) <https://lucid.app/documents/view/c6c65f1a-d09a-422a-ade8-80eab27f8dcb>

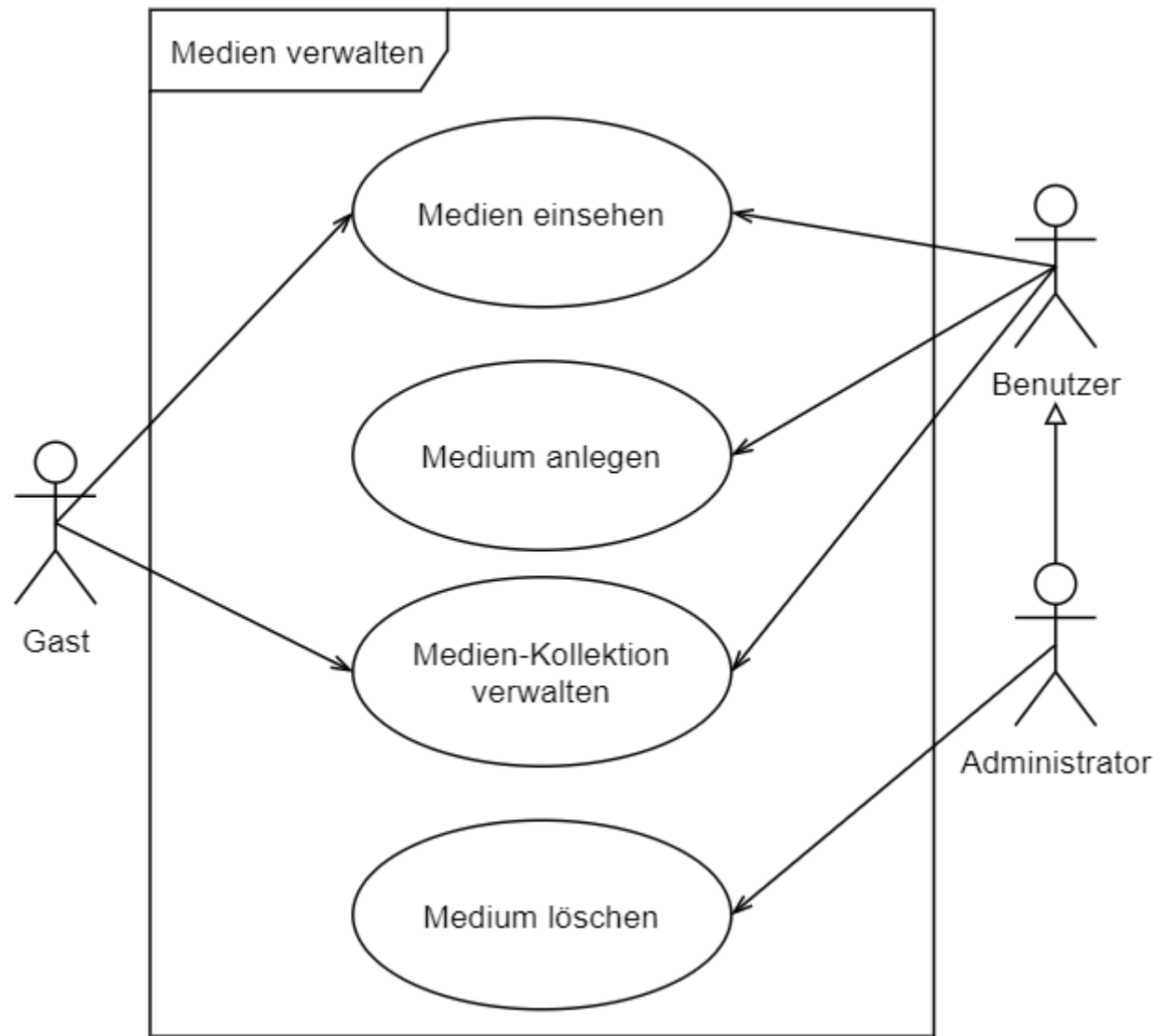
UML-Klassen Diagramm (Java02)



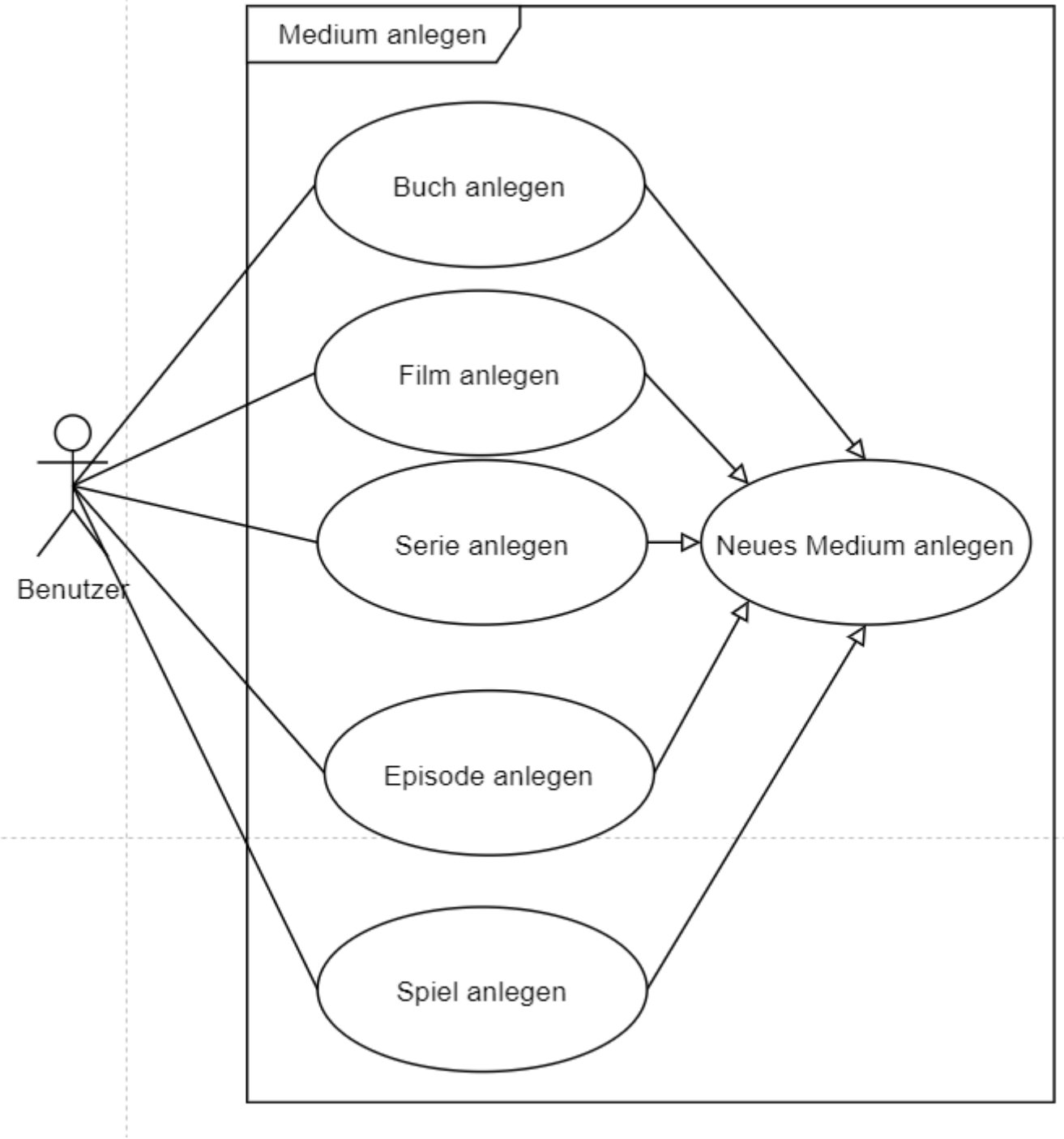
Business-Use-Case

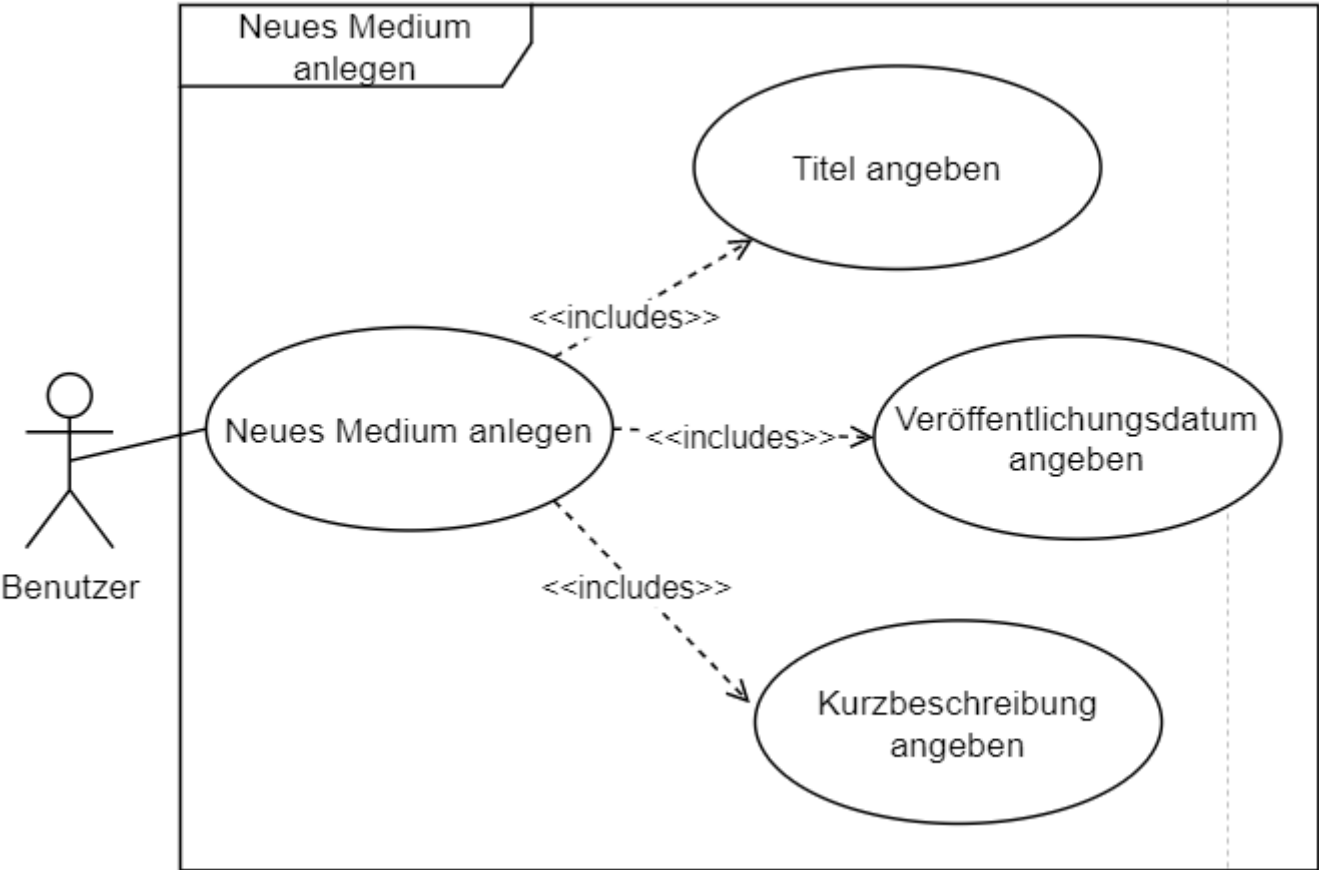


Medien verwalten

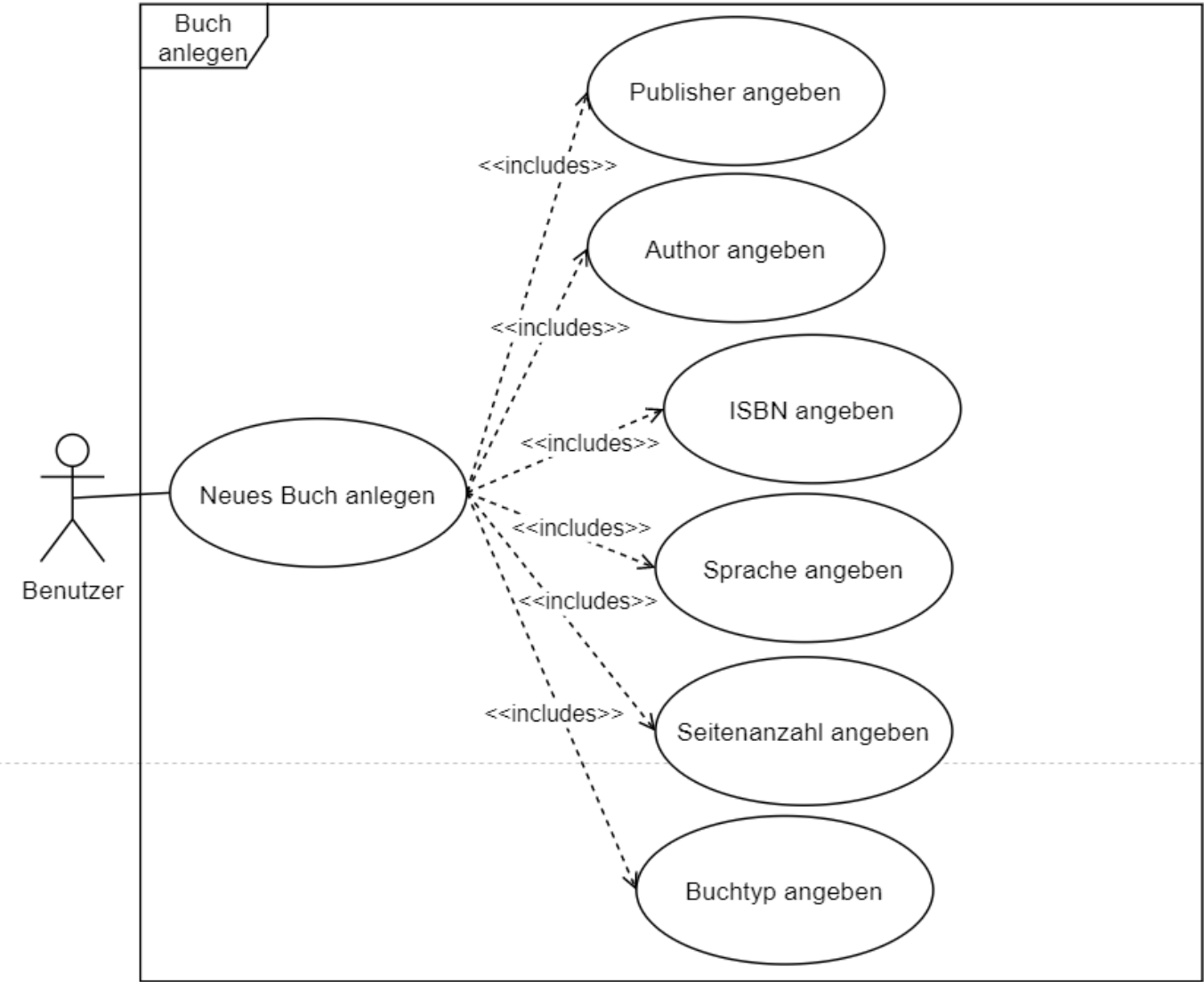


Medium anlegen

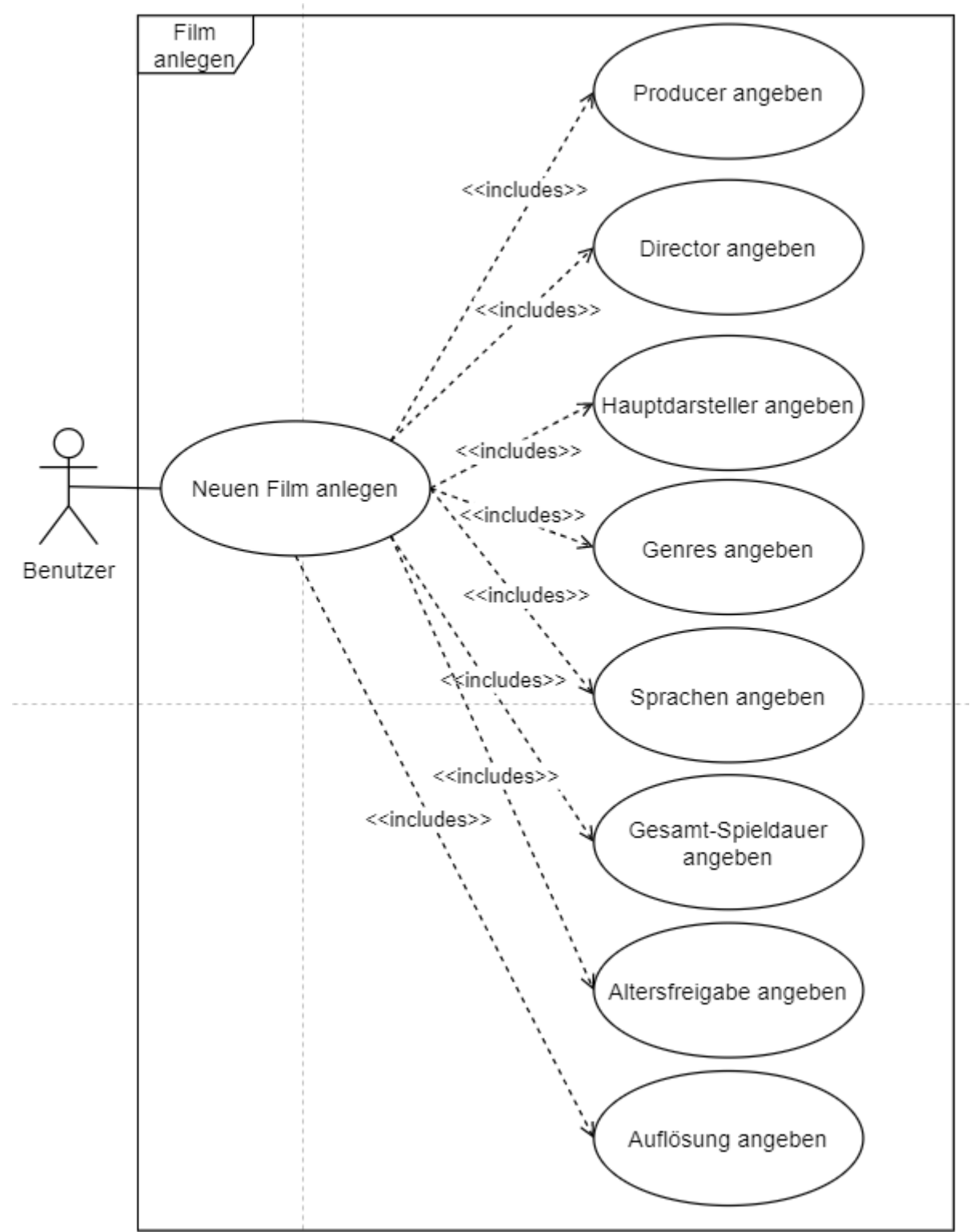




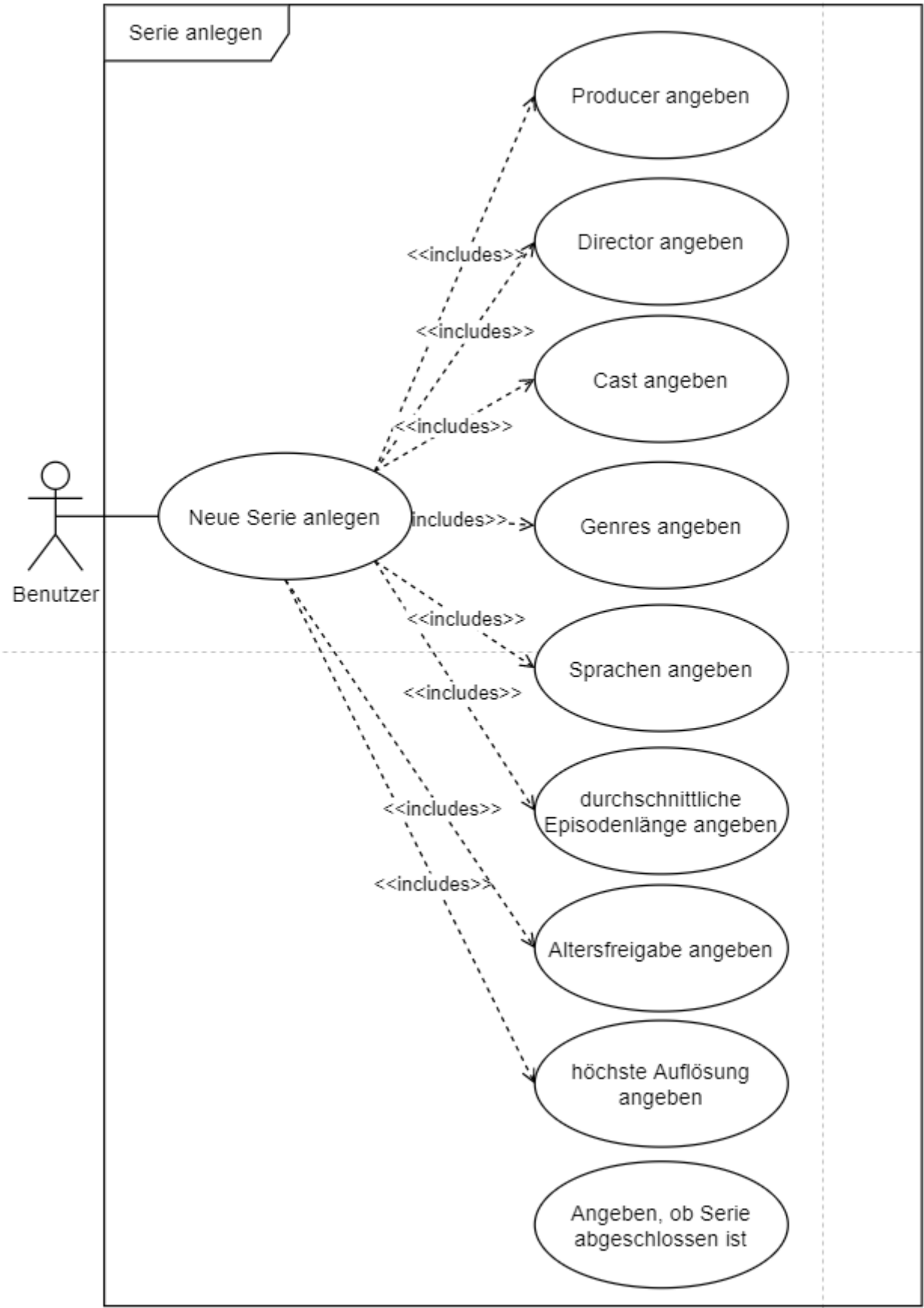
neues Buch anlegen



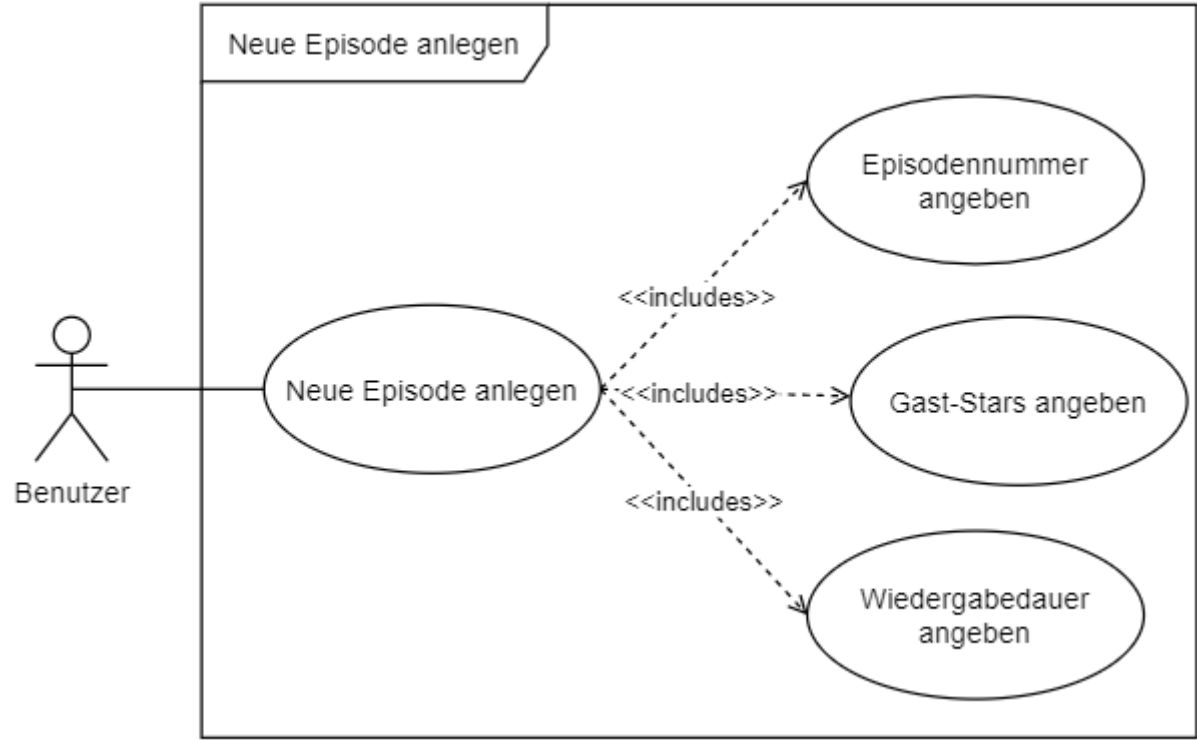
neuen Film anlegen



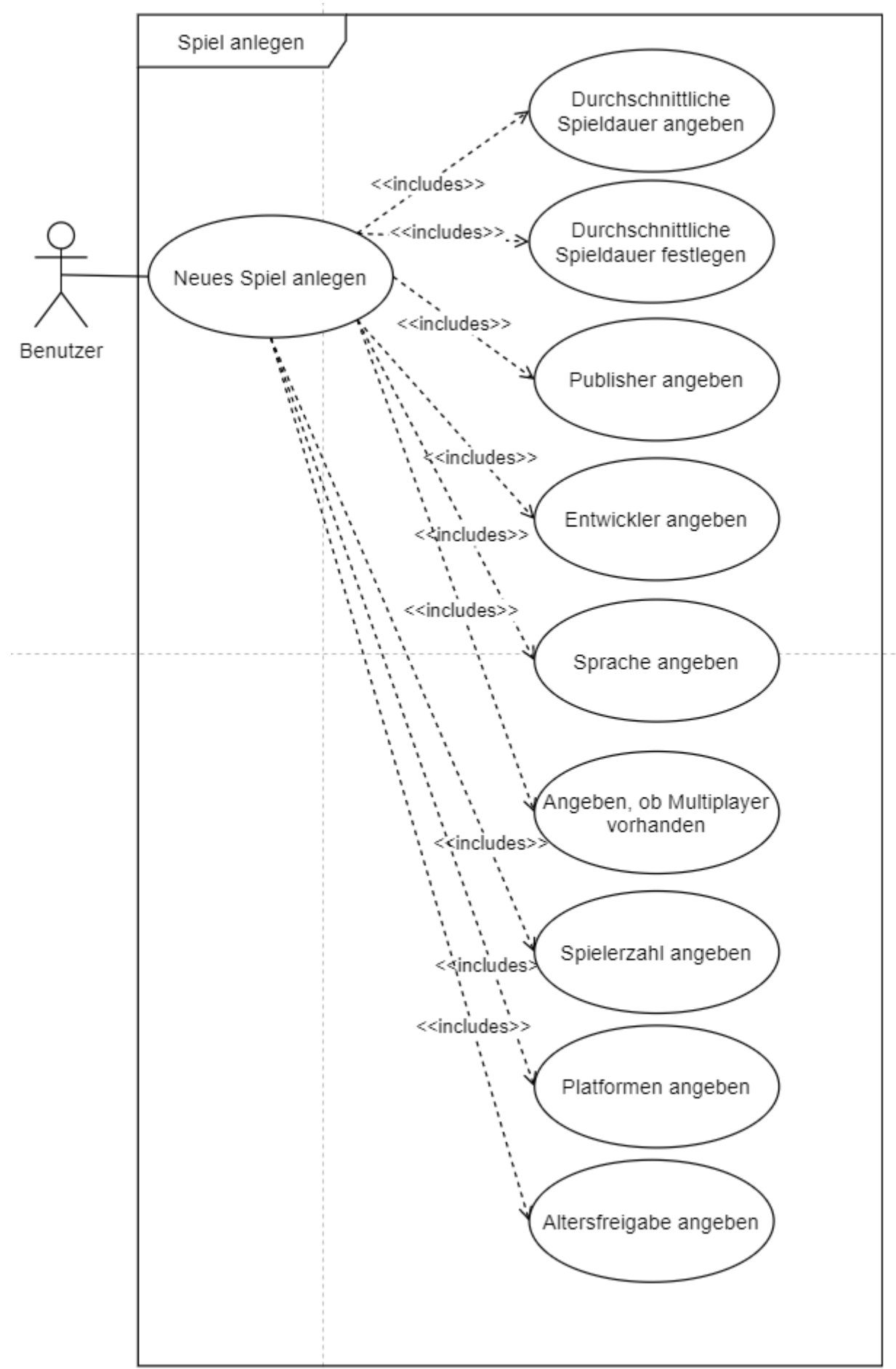
neue Serie anlegen



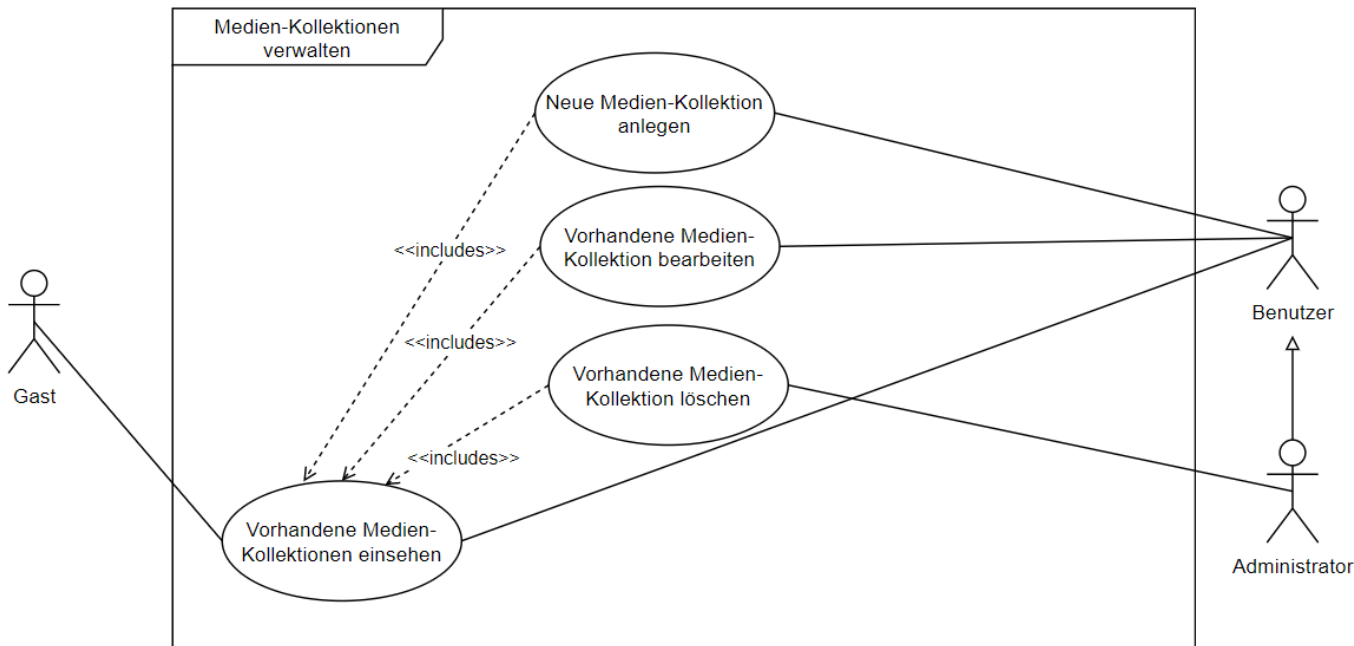
neue Episode anlegen



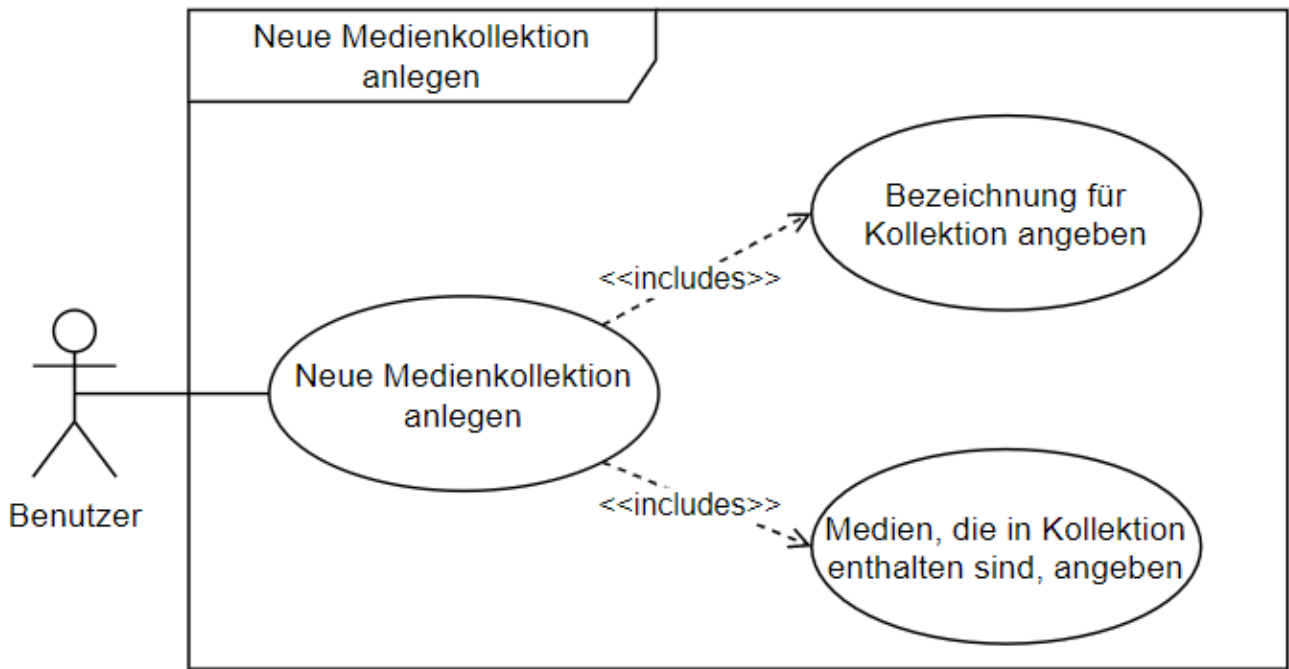
neues Spiel anlegen



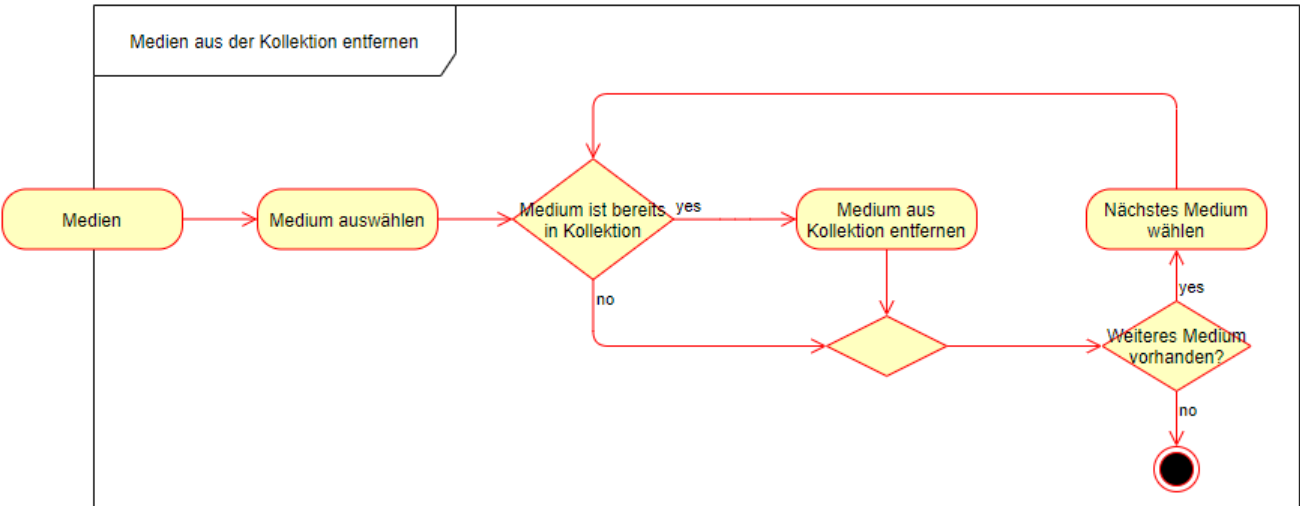
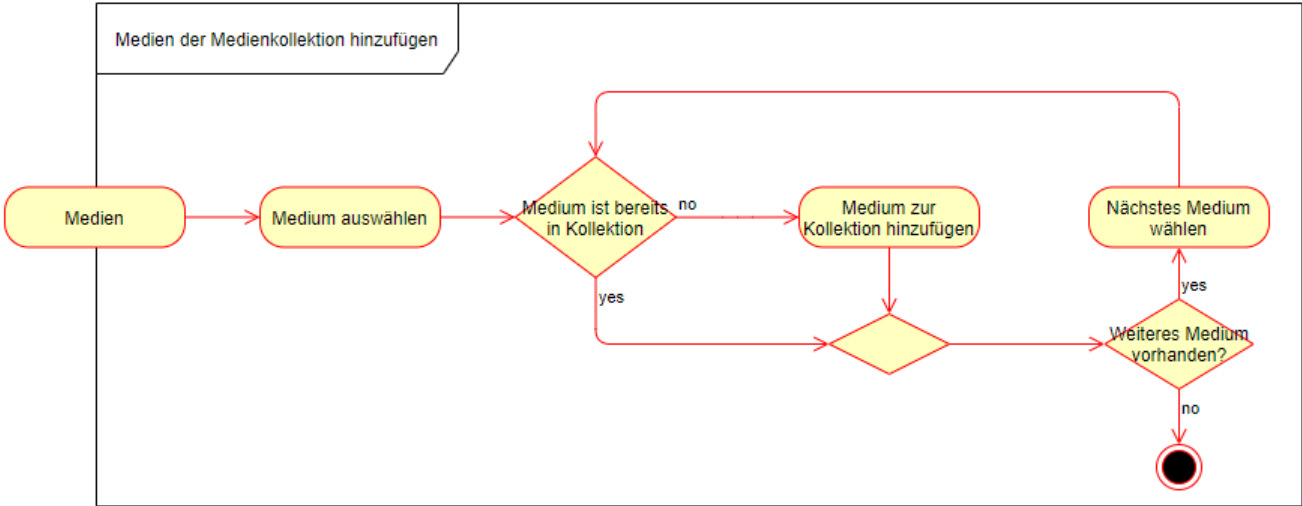
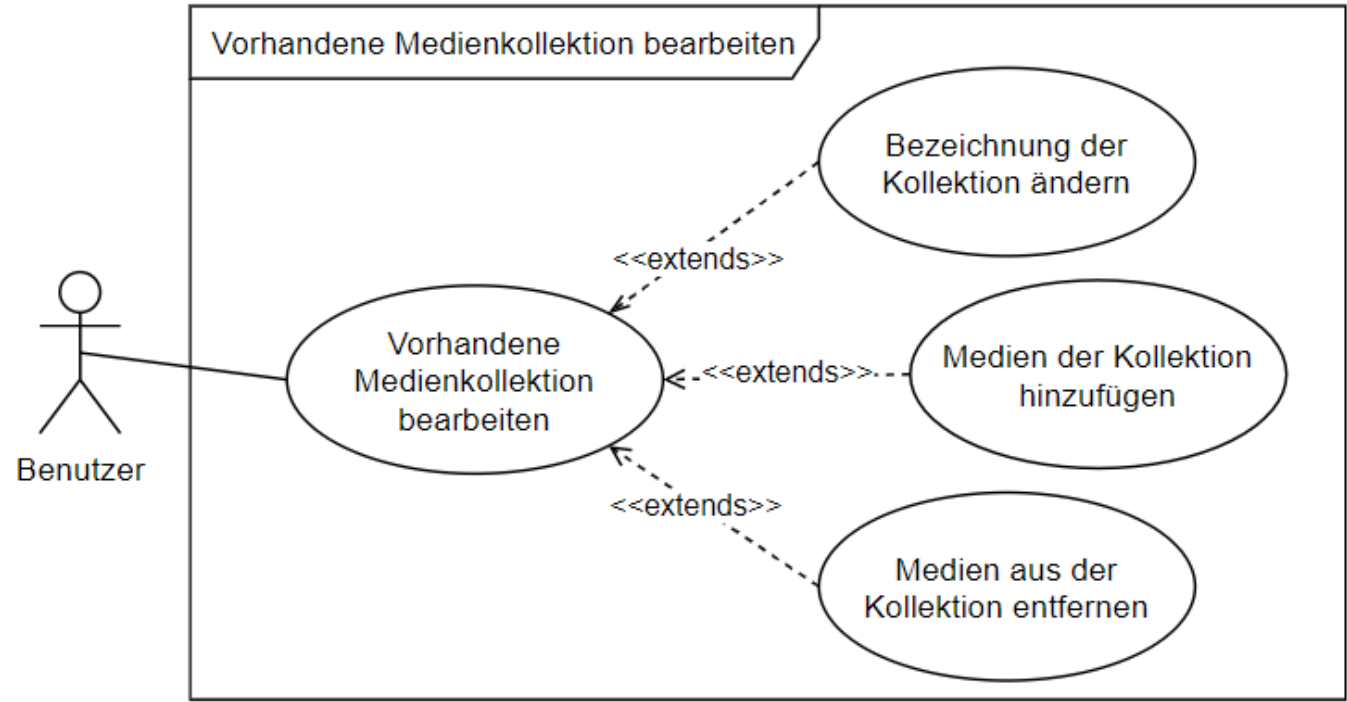
Medien-Kollektion verwalten



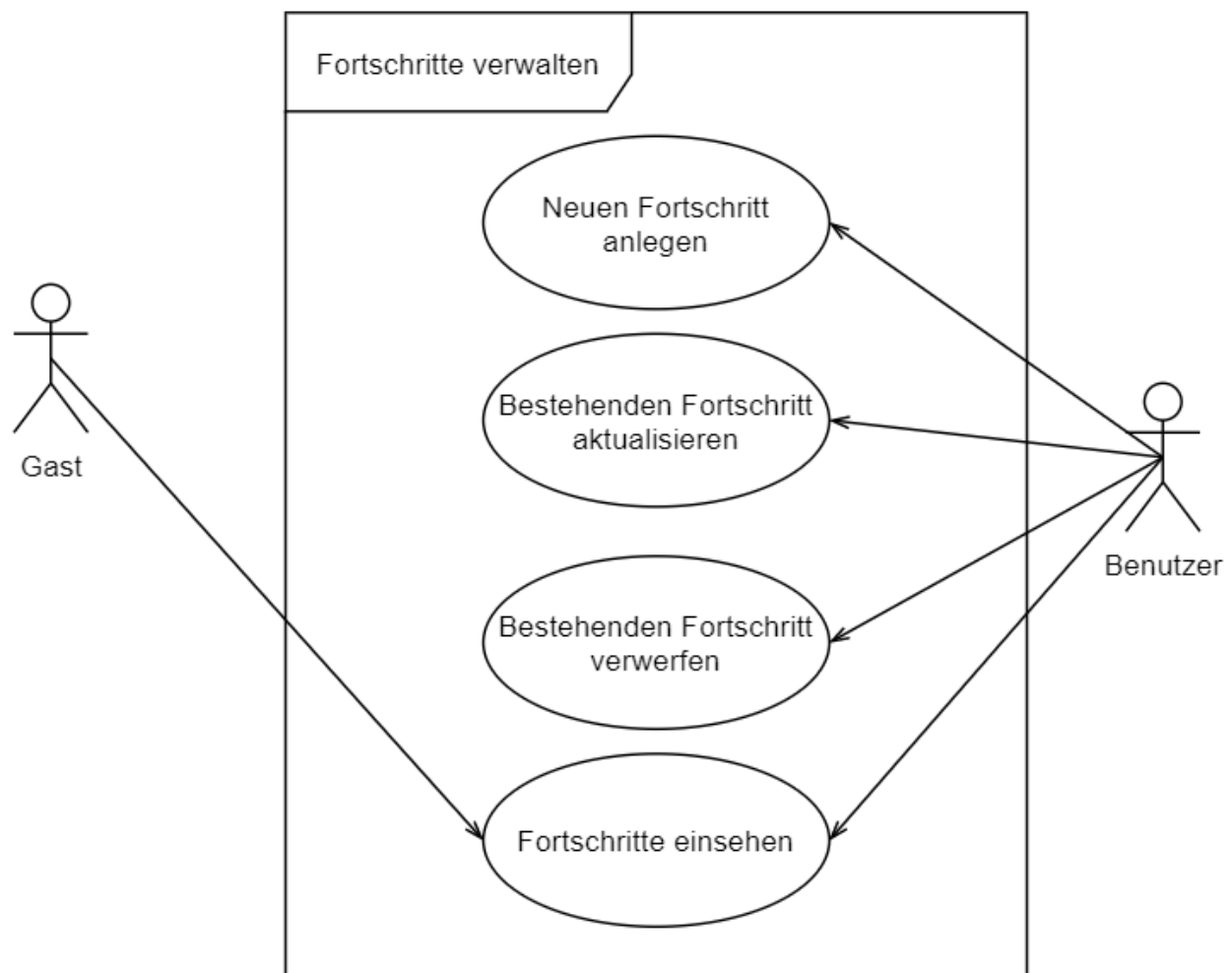
Medien-Kollektion anlegen



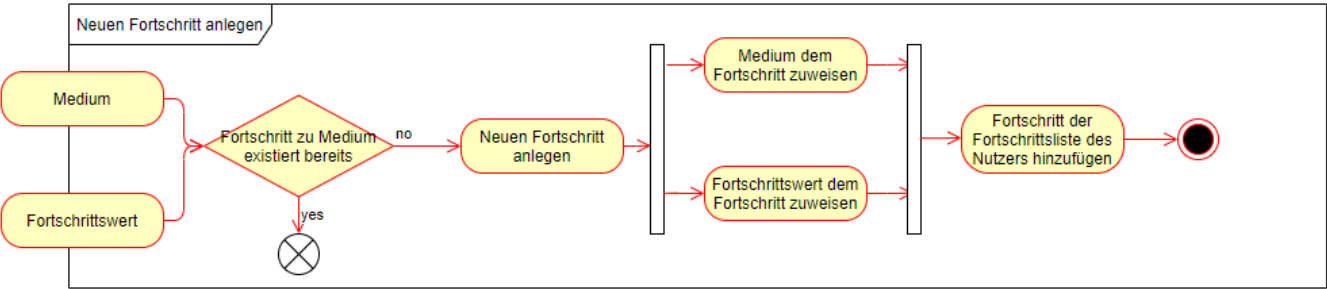
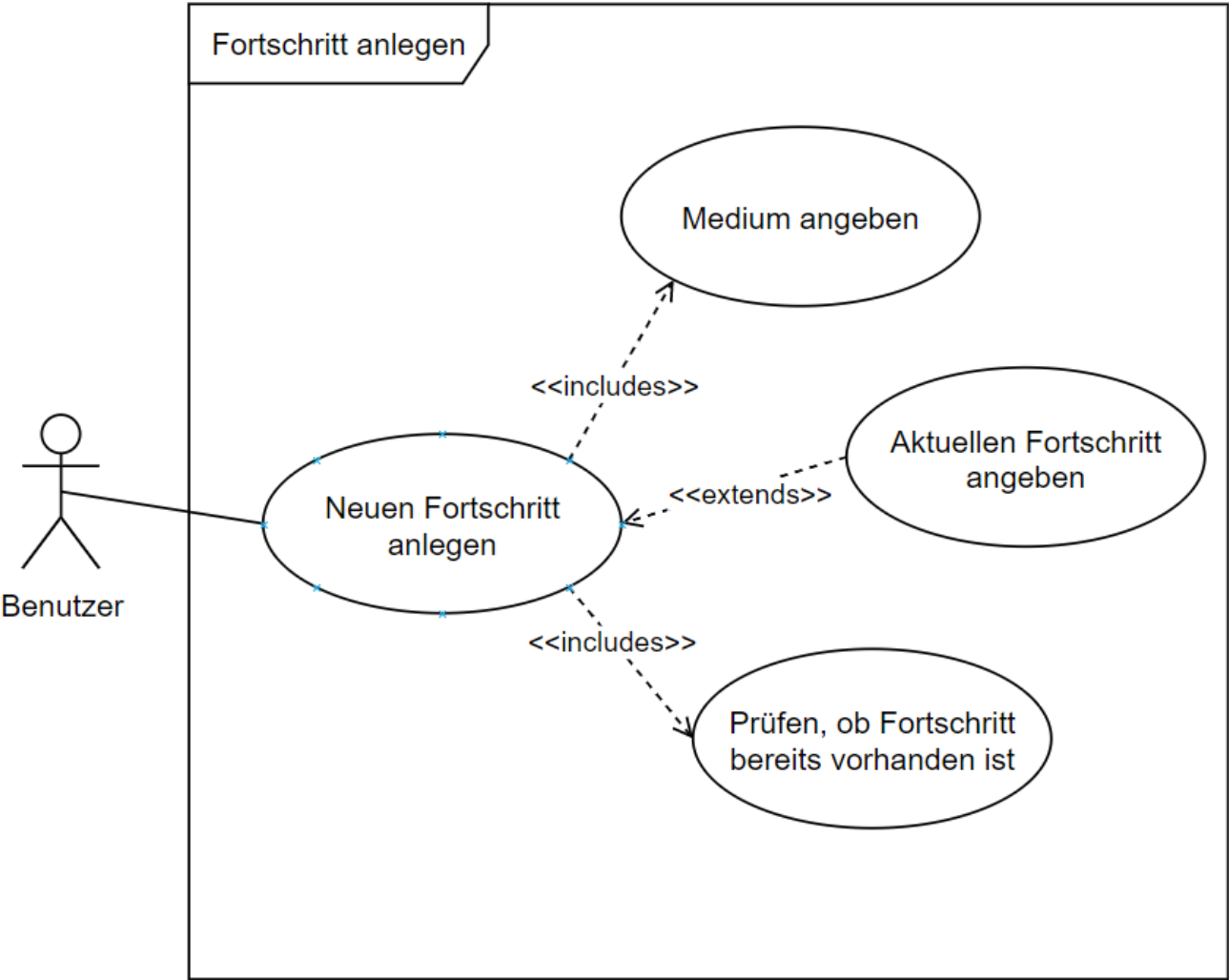
vohandene Medien-Kollektion bearbeiten



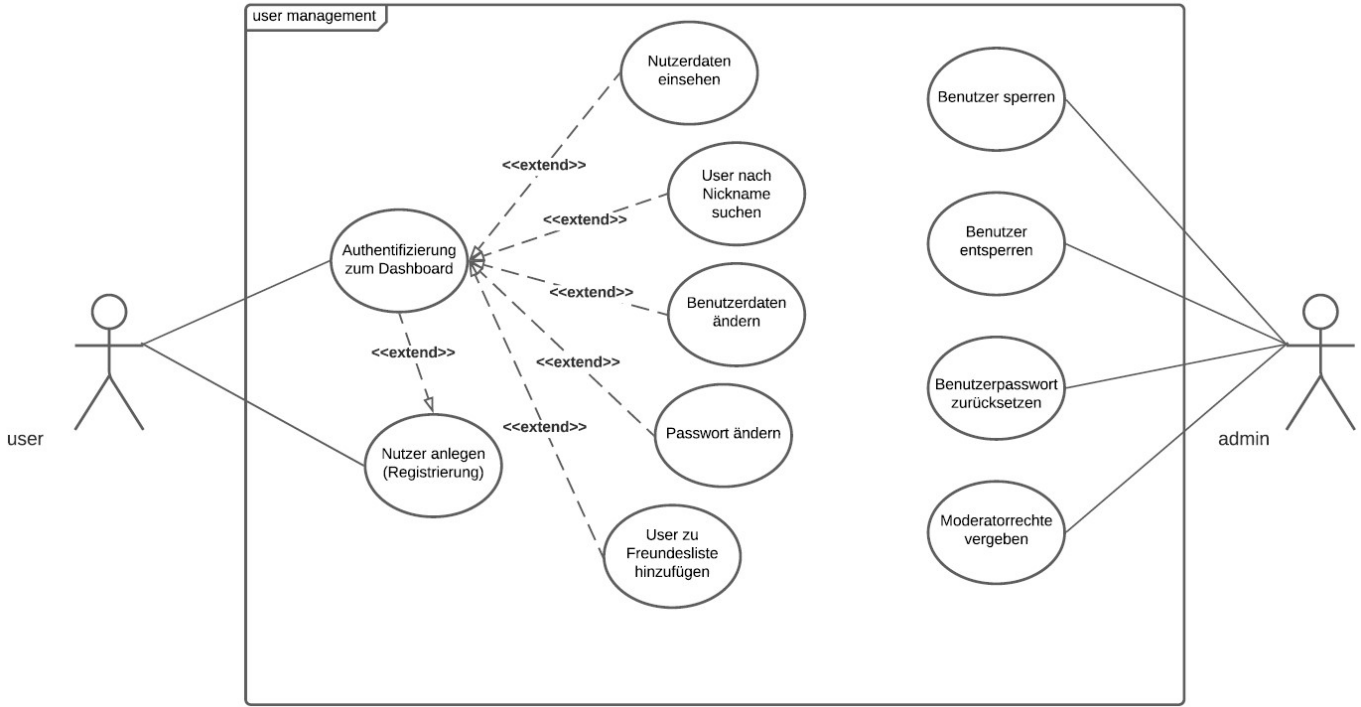
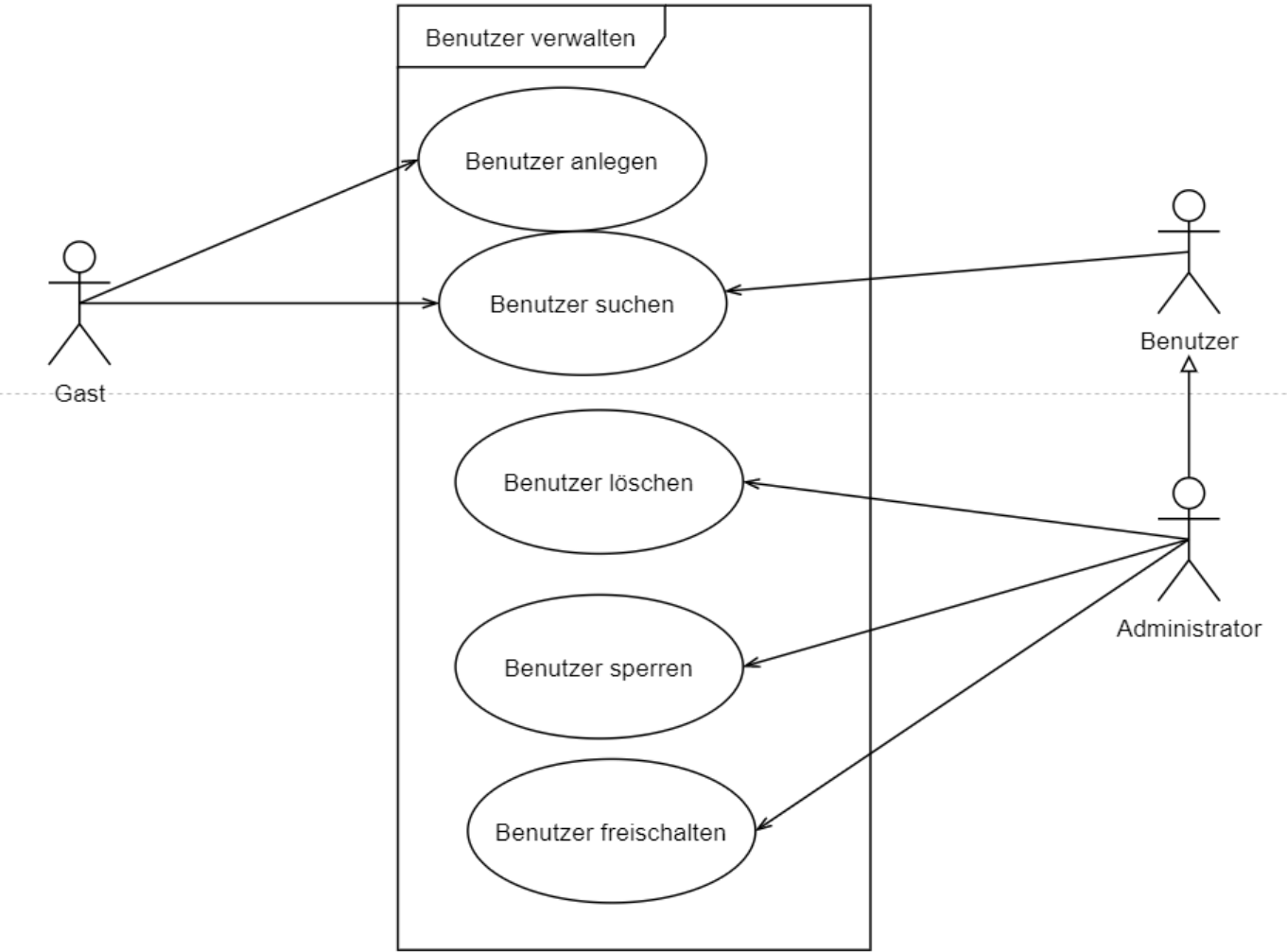
Fortschritte verwalten (Java01)

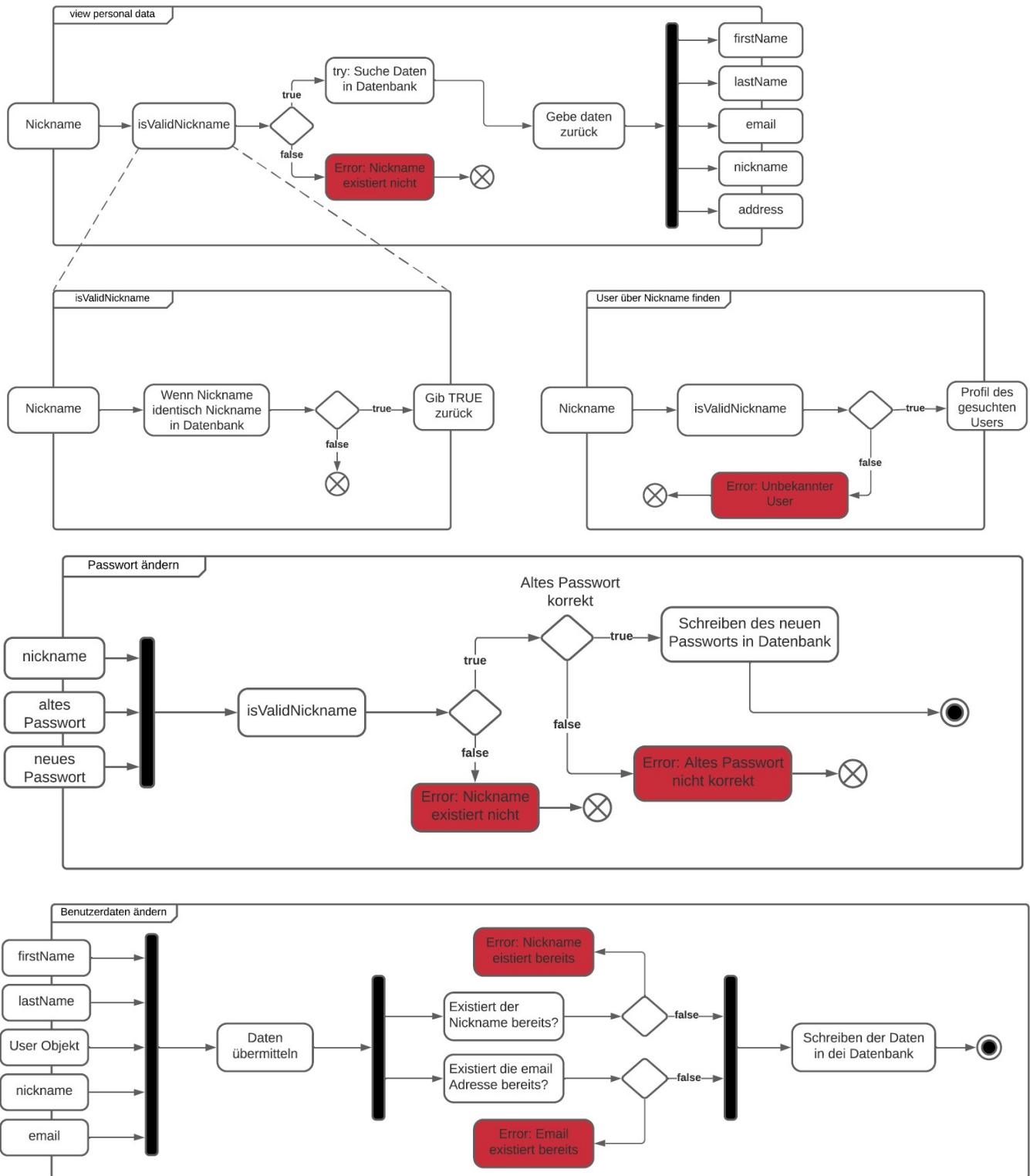


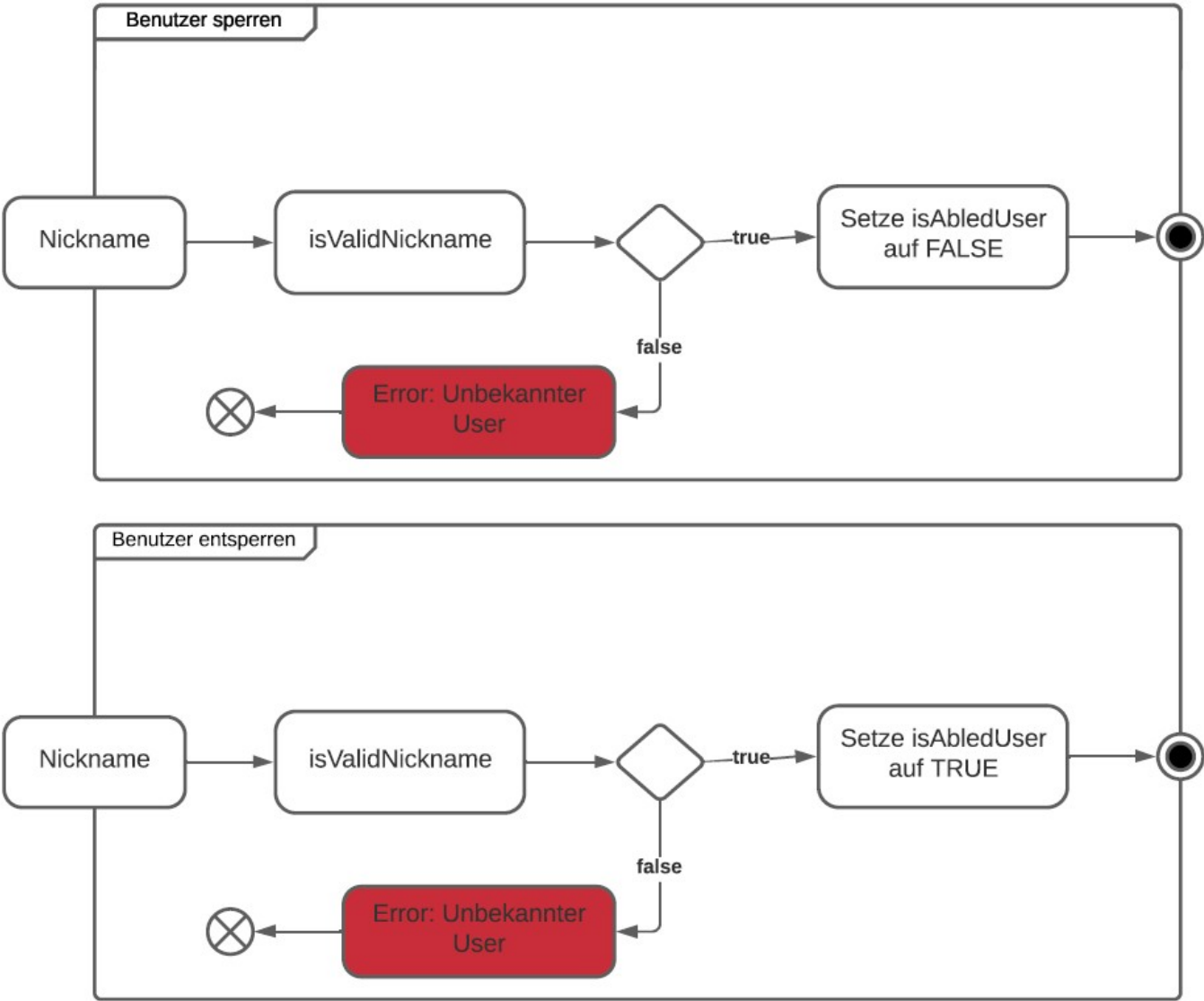
Neuen Fortschritt anlegen (Java01)

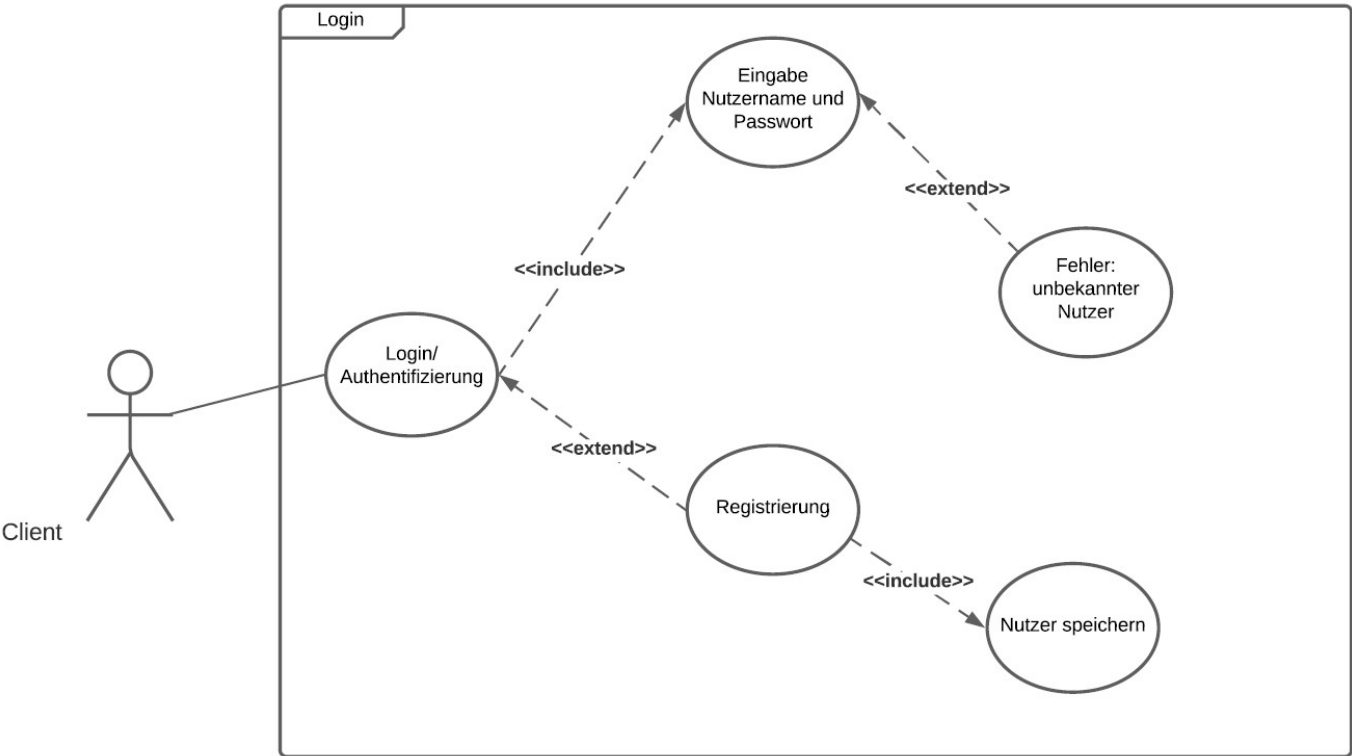
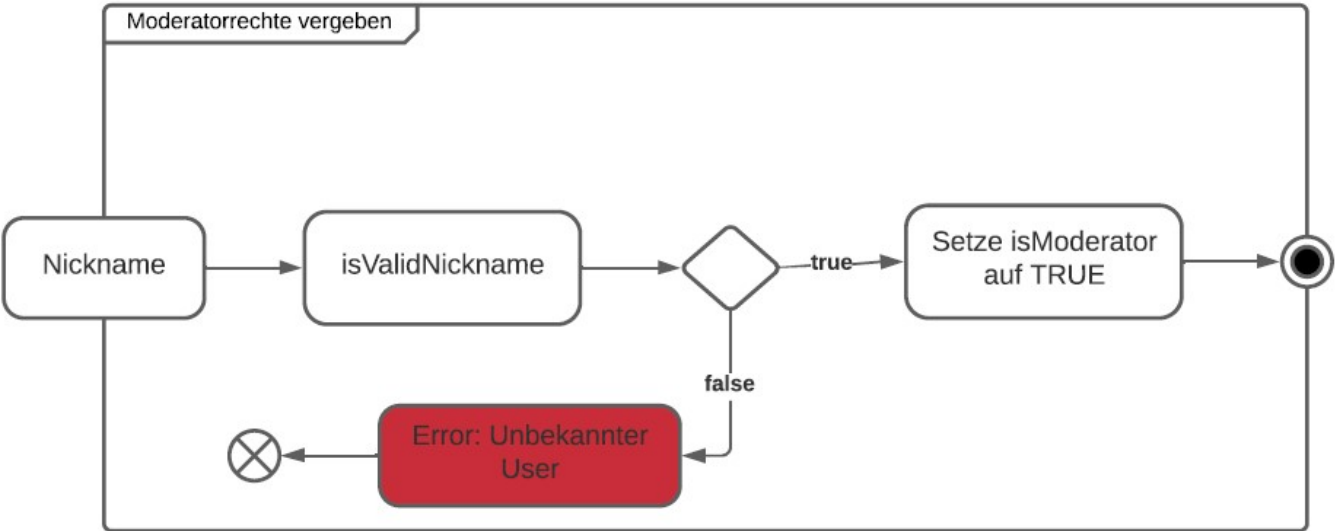
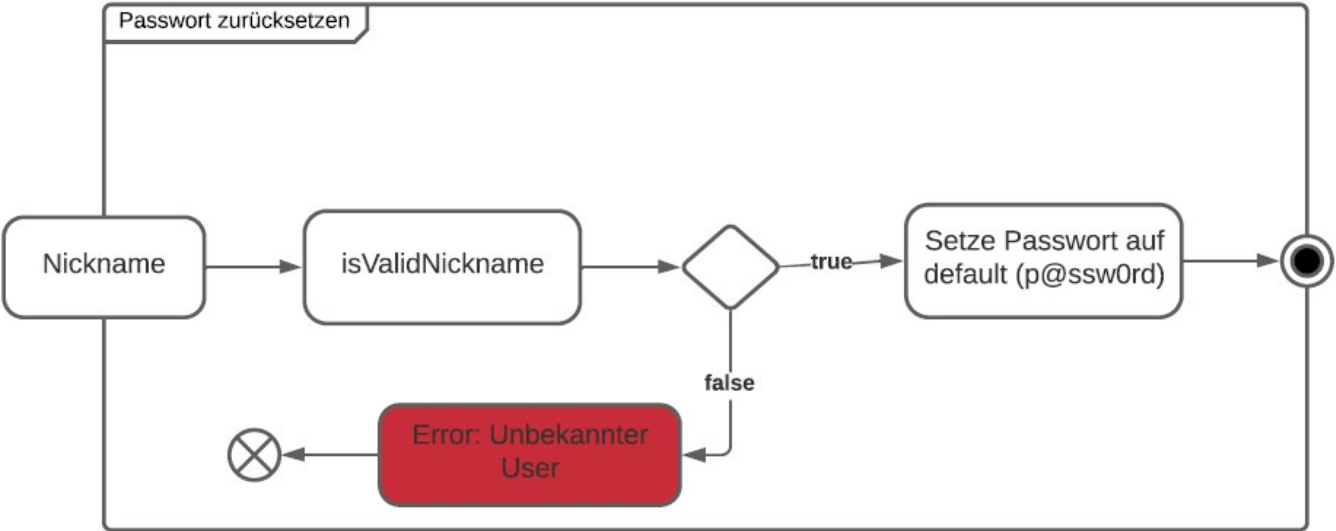


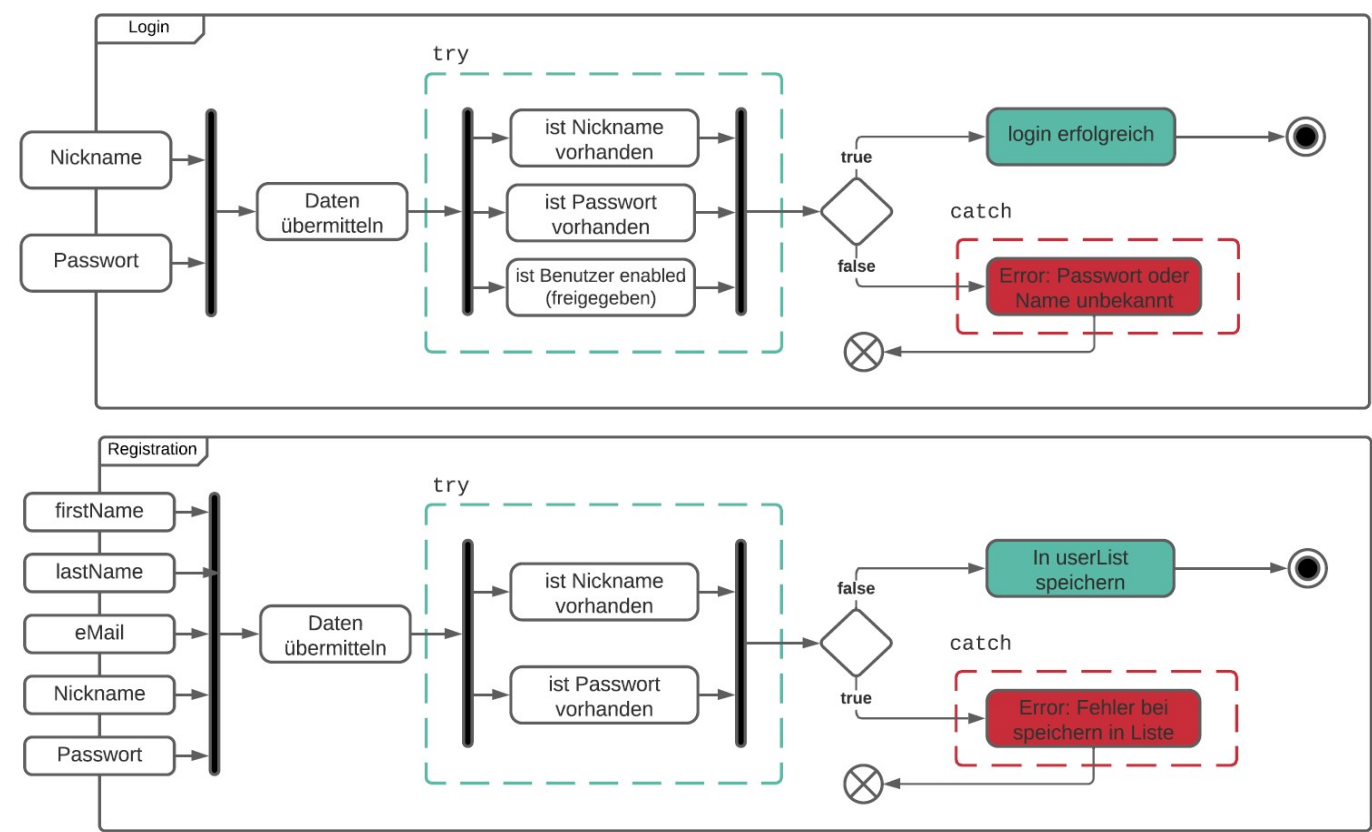
Benutzer verwalten



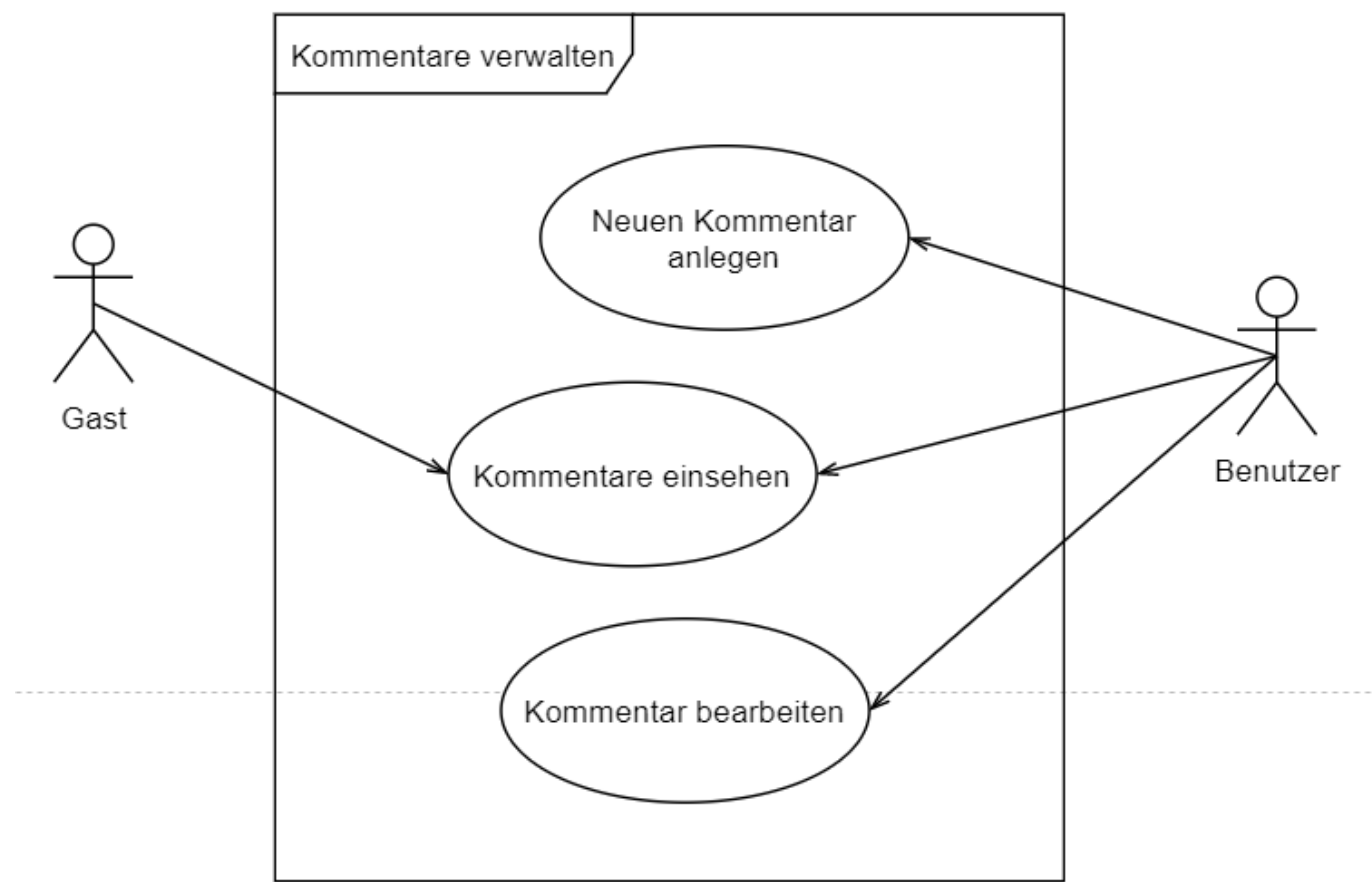


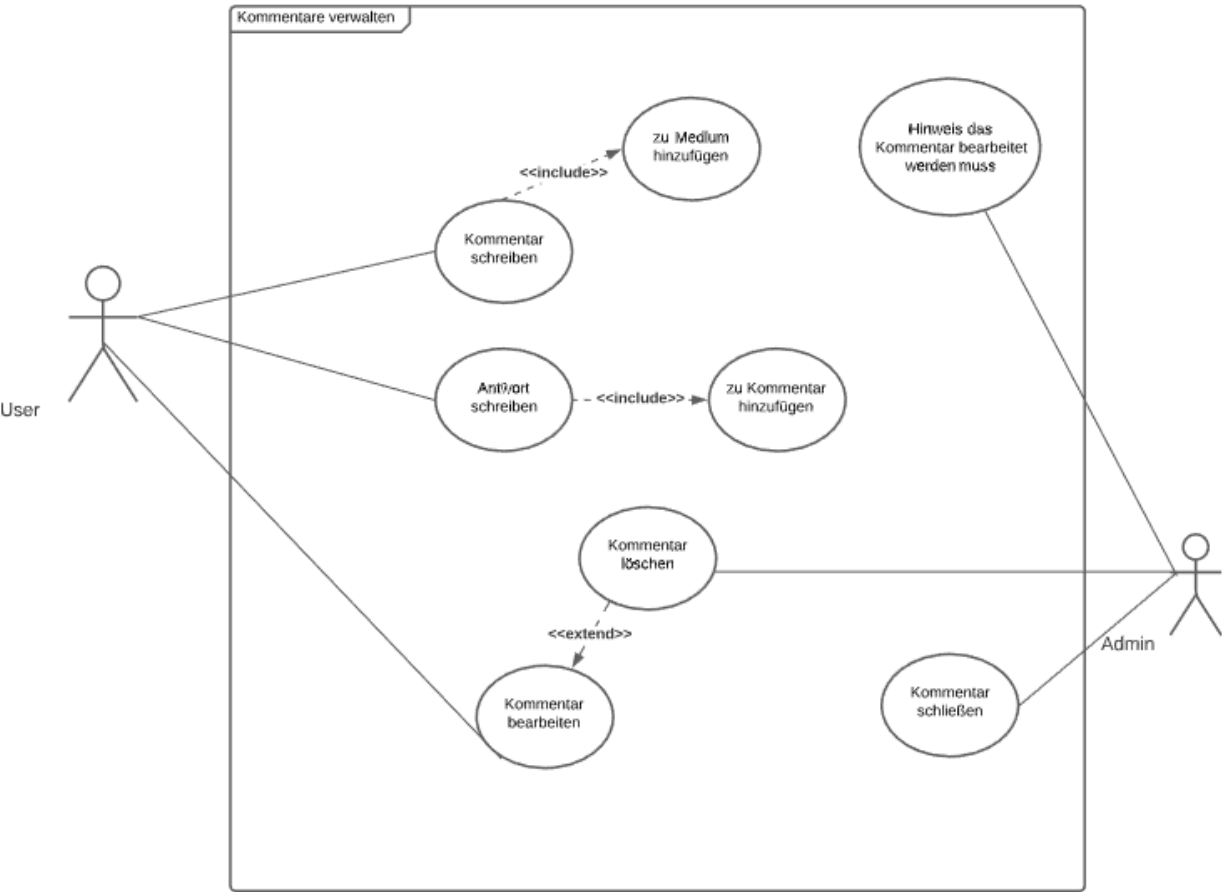


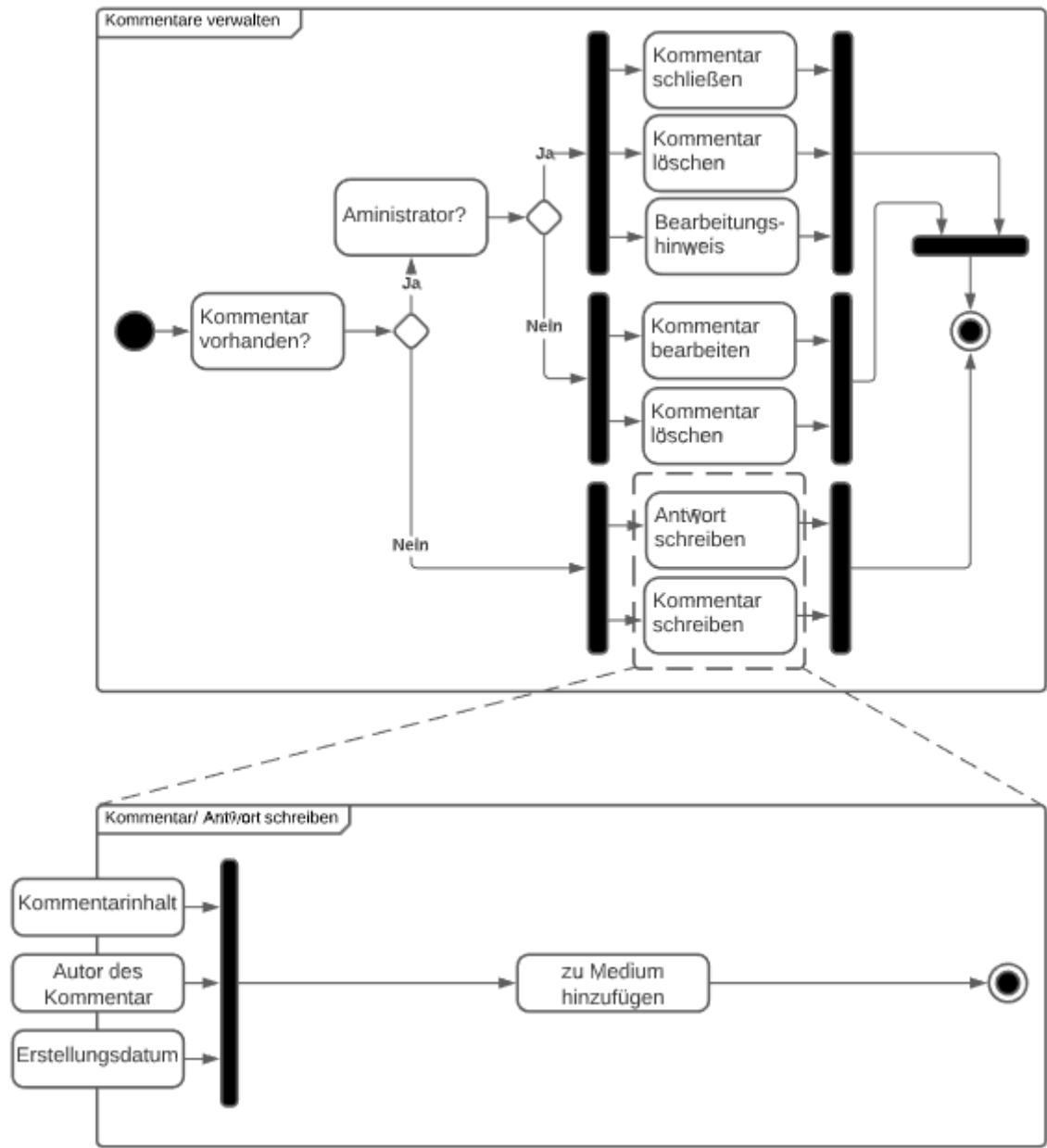




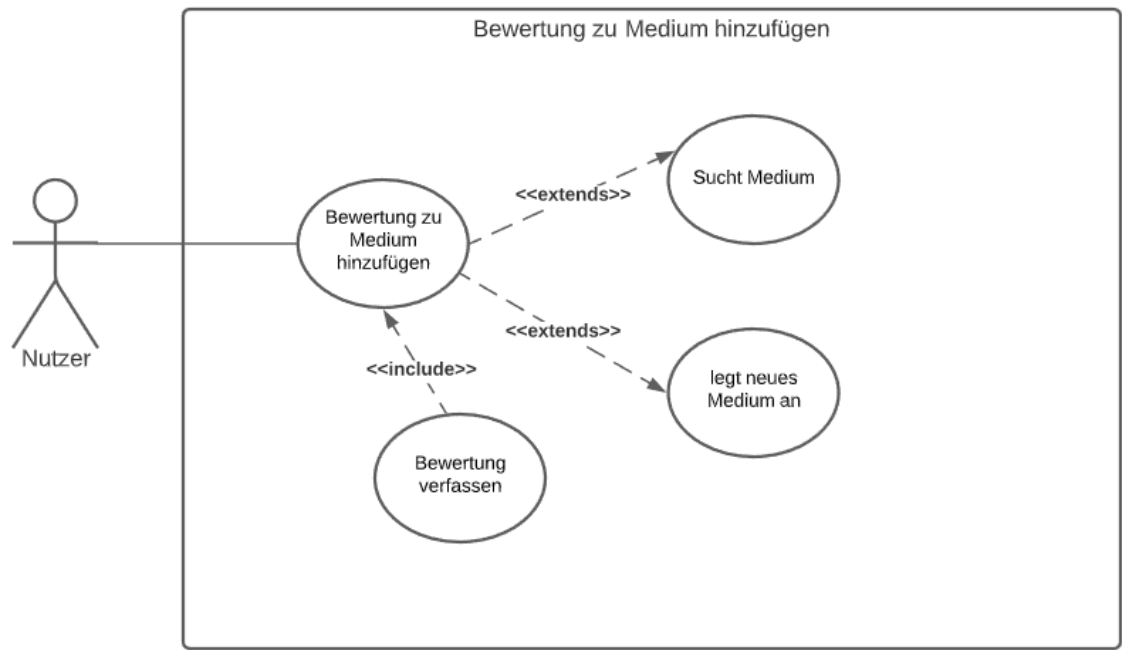
Kommentare verwalten



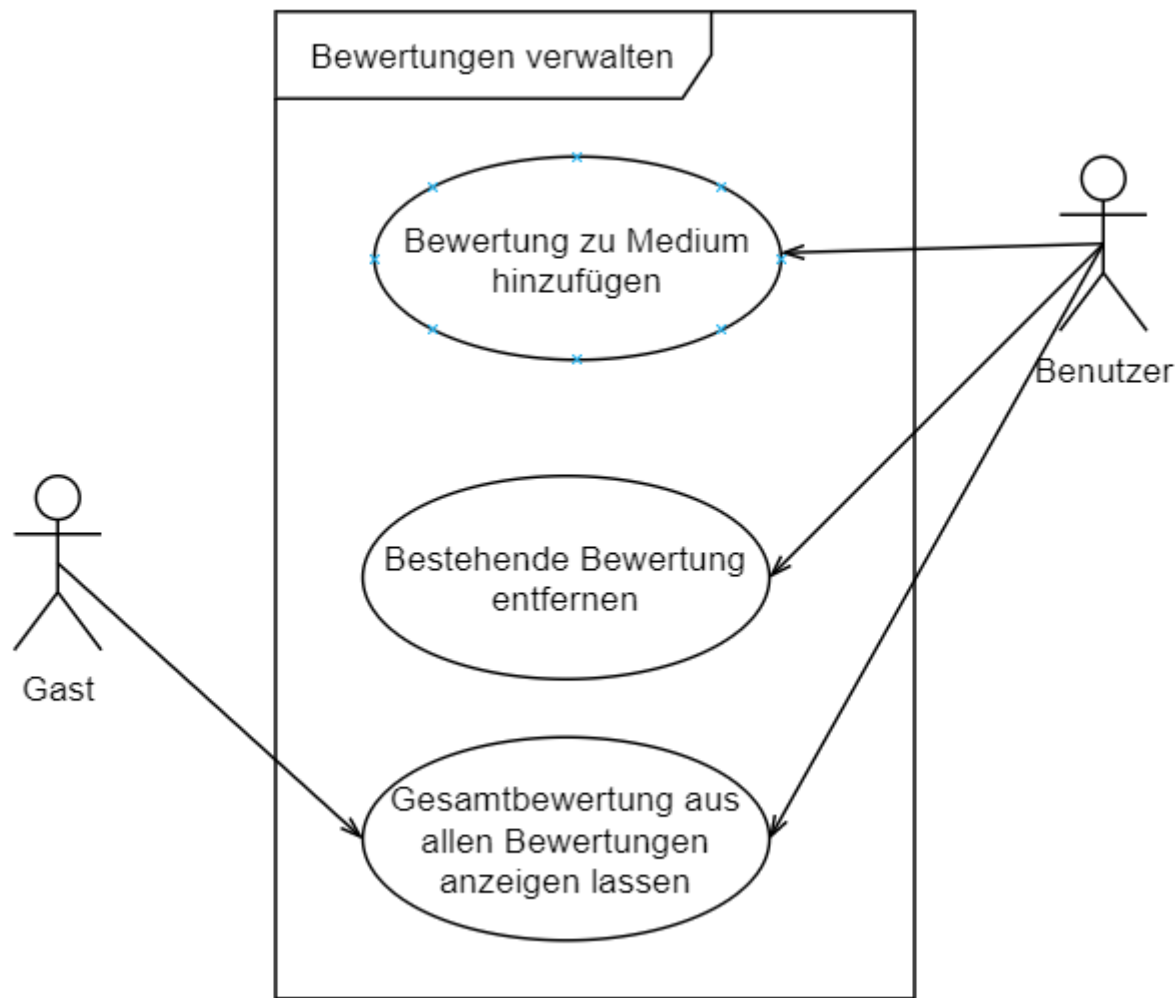


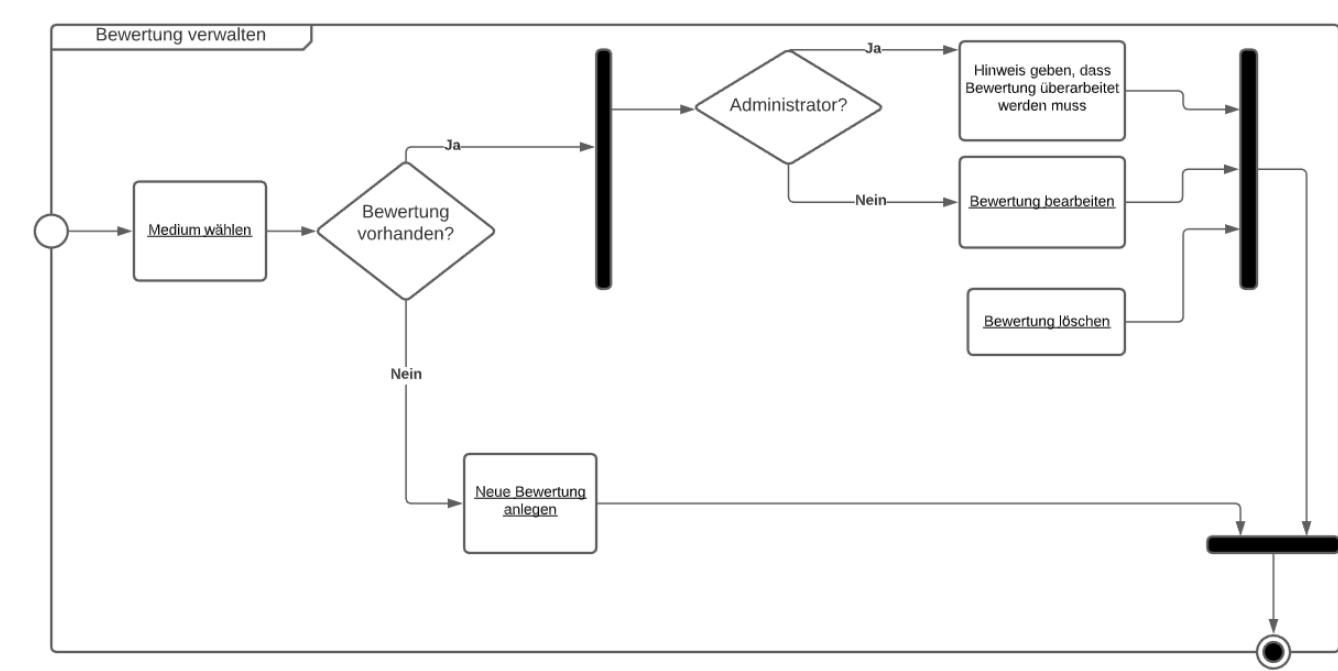


Bewertungen zu Medium hinzufügen

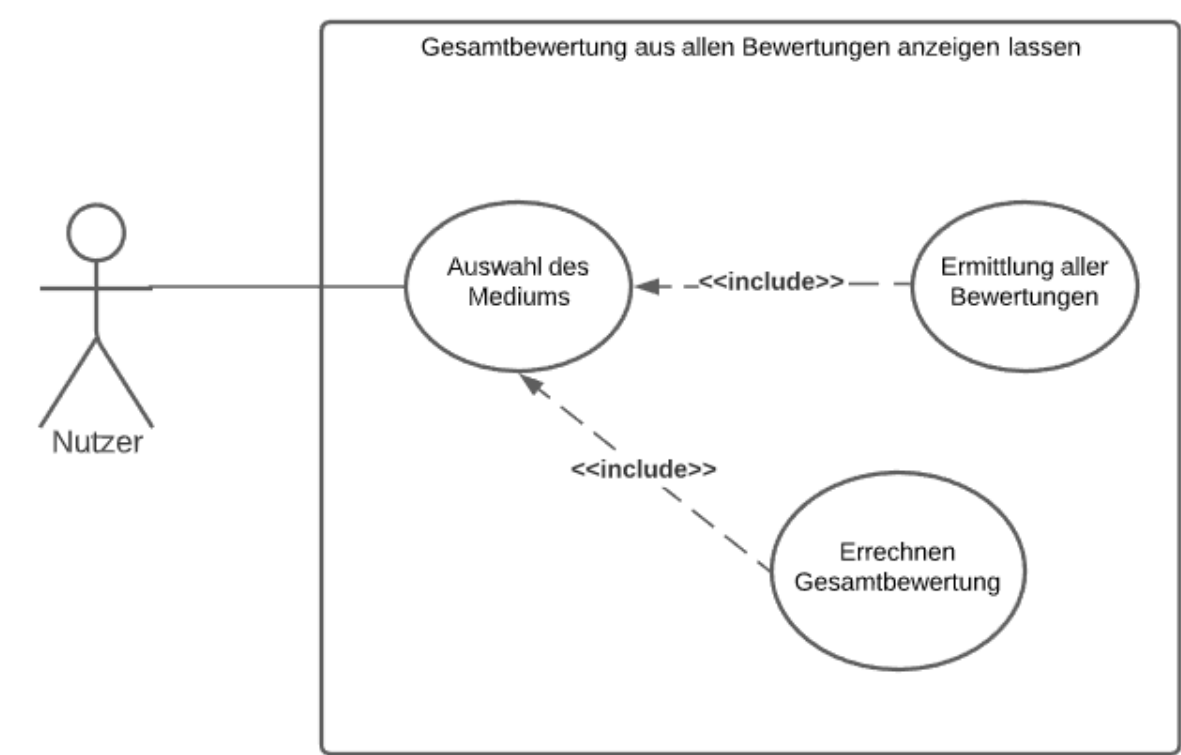


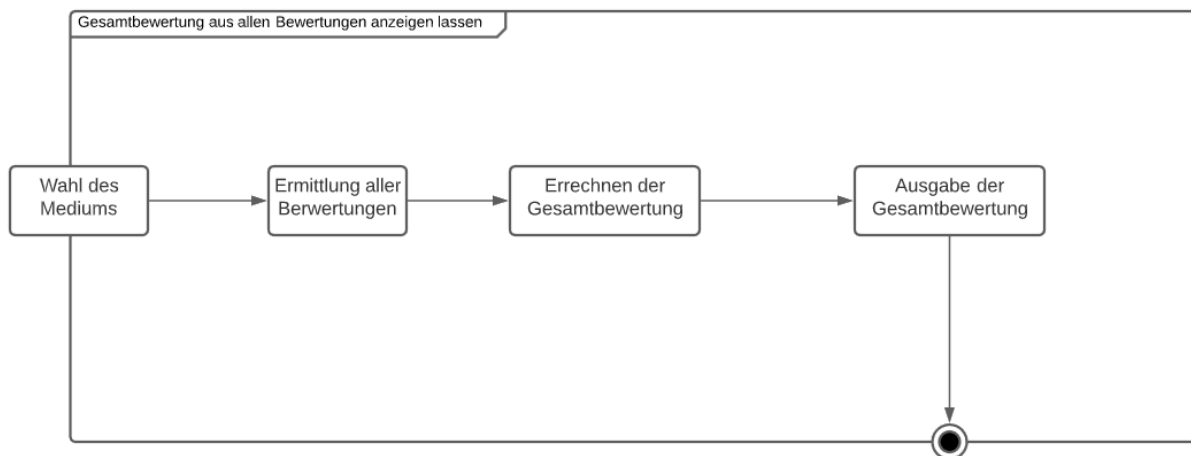
Bewertungen verwalten



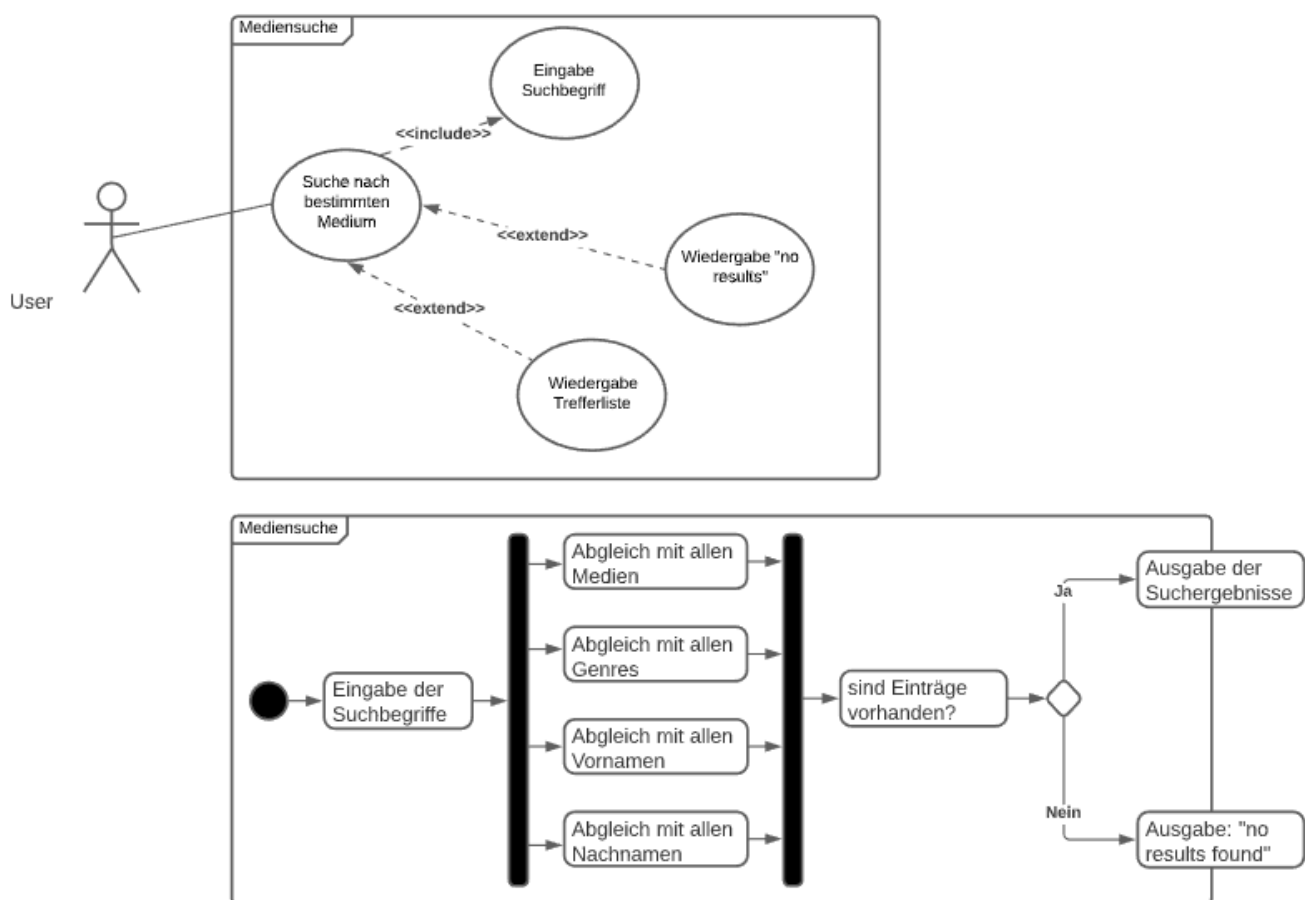


Gesamtbewertung aus allen Bewertungen anzeigen lassen





Medium suchen



Projektstruktur

Das Projekt ist grundlegend in folgende Packages untergliedert:

- controllers (beinhaltet die REST Controller sowie die Advices für geworfene Exceptions)
- core (beinhaltet grundlegende Konfigurationsdateien für z. B. Spring Security, CORS, Swagger, Madelmapper)
- dtos (Beinhaltet die Definitionen der Data Transfer Objects, die von den Rest Controllern an den Client gesandt werden)
- errors (beinhaltet alle Exceptions, die für das Projekt angelegt wurden)

- models (beinhaltet die Definitionen der Entitäten, die in die Datenbank gespeichert werden)
- repositories (beinhaltet die Interfaces zum Zugriff auf die Datenbank)
- security (beinhaltet für Spring Security erforderliche Dateien)
- services (beinhaltet verschiedene Hilfsfunktionen für die Controller)
- util (beinhaltet einen speziellen Service für Json Web Token)

In jedem der Packages wurde darauf geachtet eine gleiche Strukturierung für die enthaltenen Klassen vorzunehmen, um diese thematisch besser zu untergliedern. Im Grunde genommen ist folgende Strukturierung enthalten:

- media
 - books
 - isbn (nur vorhanden, wenn isbn dateien vorliegen)
 - collections
 - games
 - movies
 - series
 - users
 - comments (nur vorhanden, wenn mehr als eine Datei zu Kommentaren gehört)
 - ratings (nur vorhanden, wenn mehr als eine Datei zu den Kommentaren gehört)
-

Umgesetzte Funktionalitäten

Nachfolgend ist aufgelistet, welche Funktionen das im Rahmen von Java 02 fertiggestellte Projekt bietet:

- Nutzerverwaltung
 - Nutzer anlegen
 - Nutzer bearbeiten
 - Nutzer verwalten (Admin)
 - Anmeldung mithilfe von Spring Security
- Kommentarverwaltung
 - Anlegen von Kommentaren
 - Bearbeiten von Kommentaren
 - Löschen von Kommentaren
 - Antworten auf Kommentare
- Medienverwaltung
 - Anlegen von verschiedenen Medien (Buch, Film, Spiel, Serie, Episode)
 - Bearbeiten von Medien
 - Suchen nach Medien
 - Hinzufügen von Medien zu einer Sammlung/ Collection (Hierfür muss unter "Meine Sammlungen" eine Sammlung erstellt werden. Anschließend kann in der Detailansicht eines Mediums ein Medium einer Sammlung hinzugefügt werden)
 - Anzeigen von Medien eines bestimmten Medientypes
- Bewertungsverwaltung
 - Hinzufügen von Bewertungen (Jeder Nutzer darf nur eine einzige Bewertung pro Medium abgeben, kann diese aber ggf. noch bearbeiten)
 - Bearbeiten von Bewertungen

- Löschen von Bewertungen

Das Frontend wurde mithilfe von ReactJS realisiert.

Genutzte Tools

IDEs

- IntelliJ
- VSCode
- WebStorm

Wissenserwerb

- YouTube
- Stackoverflow
- Baeldung

Sonstiges

- Spring Boot
- Spring Boot Security
- React.js
- SpringFox / Swagger
- @Material-UI
- Xampp (Apache / mariaDB)
- Discord
- JUnit
- GitHub

Zusammenarbeit

Im Rahmen des Projektes haben wir auf gewissen Elemente von Scrum und Kanban zurückgegriffen.

Das Team um "RateYourStuff" haben sich alle 7 Tage zusammengefunden. Zu diesen Treffen hat jeder seinen PPP (Progress, Planned, Problems) der Runde mitgeteilt und es wurden die Ziele für die nächsten 7 Tage gesetzt. Diese gesetzten Ziele wurden in einem Kanban-Board festgehalten.

Kanban-Board des Teams <https://github.com/fh-erfurt/RateYourStuff2.0/projects/1>

Das Board wurde unterteilt in

"ToDo" --> Hier wurden alle Issues eingestellt, die erledigt werden mussten

"In progress 'Vorname des Teammitgliedes'" --> Jedes Teammitglied konnte sich dann selbst Issues aus dem Bereich ToDo nehmen und in seinen Bereich verschieben. So konnte jeder aus dem Team sehen wer sich gerade um welchen Part kümmert. Issues wurden meistens mit Checklisten versehen, so konnte auch gleichzeitig mit eingesehen werden, wie weit der Issue abgearbeitet ist

"currently Done" --> in diesem Bereich des Boards wandern alle Issues, die fertiggestellt wurden. dort verweilen sie bis zum nächsten Meeting. Bei diesem Meeting wird über die fertiggestellten Issues gesprochen und wenn

diese in Ordnung sind, werden sie geschlossen und wandern in den Bereich "Done".

"Done" --> in diesem Bereich befinden sich alle geschlossenen Issues, also alles was bisher abgearbeitet und für "fertig" erklärt wurde.

Die Aufgaben wurden somit fair untereinander aufgeteilt, jeder hat meistens das genommen was ihm am besten gelegen hat und am Ende wurde alles durch das Team geprüft.

Alles in allem lief das Projekt nahezu problemlos ab. Was auf unsere gute Planung und unser sehr gut harmonisierendes Team zurückzuführen ist

Lessons Learned

Wie schon im vorhergehenden Part erwähnt hatten wir keine wirklichen Probleme in unserem Projekt. Die meisten Probleme existierten durch Unwissenheit und mit einem geringen Aufwand an Selbststudium konnten diese auch behoben werden. Spätestens durch gegenseitige Hilfe waren Probleme schnell vom Tisch.

Dennoch soll dieses Kapitel nicht gänzlich leer bleiben.

- Listen die in einer Klasse immer wieder benutzt werden, sollten als Attribut dieser Klasse verwendet werden!
- Keep it simple / stupid --> Kleine Funktionen schreiben, die in komplexeren Methoden zusammen verwendet werden können!