

Anforderungsdokument

1. Stakeholder

Der Medianizer ist für verschiedene Endbenutzer einsetzbar. Zum einen ist er für Privatpersonen geeignet, die eine Unterstützung bei der Organisation ihrer Film- und/oder CD-Sammlung wünschen. Dies ist umso relevanter, je größer die Sammlung ist. Darüber hinaus eignet sich die Anwendung auch für kleinere Unternehmen, die mit Medien handeln. Besonders relevant sind hier Bereiche wie Second Hand oder An- und Verkauf, da hier i.d.R. keine andere Übersicht, wie etwa ein Sortiment besteht. Für alle Endbenutzer gilt, dass sie eine intuitive Bedienung, eine klare Darstellung der Daten und eine zuverlässige Speicherung ohne online-Anwendungen erwarten.

2. Funktionale Anforderungen

Im Folgenden werden zur Ermittlung der funktionalen Anforderungen drei User Stories (US) formuliert:

- US1: Als Benutzer möchte ich Filme mit Titel und CDs mit Titel und Interpret suchen können, um in einem neuen Fenster eine übersichtliche Ausgabe der vorhandenen Einträge in Tabellenform mit den Spalten Titel, ggf. Interpret (bei CDs) und Anzahl zu erhalten.
- US2: Als Benutzer möchte ich einen Eintrag in der Tabelle auswählen und ihn löschen oder die Felder Titel, Anzahl und ggf. Interpret (bei CDs) bearbeiten können.
- US3: Als Benutzer möchte ich, wenn bei der Suche keine Ergebnisse für eine Ausgabe gefunden wurden, direkt an ein Fenster weitergeleitet werden, das meine Eingaben übernimmt und mir die Möglichkeit bietet diese um die Anzahl meiner Exemplare zu erweitern und meinen Eintrag zu speichern.

Aus den User Stories (US1 bis US3) lässt sich folgendes UML-Use-Case-Diagramm, welches mit draw.io erstellt wurde, ableiten:

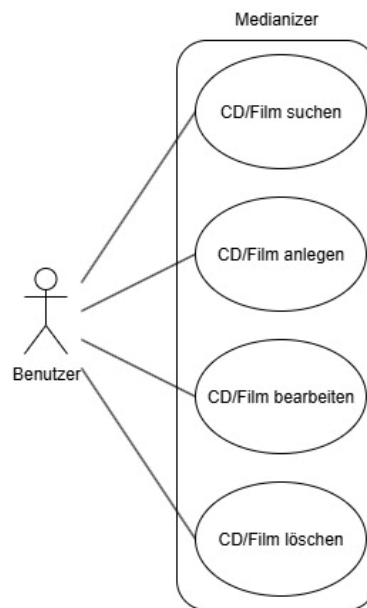


Abbildung 1

Aus den User Stories (US1 bis US3) und dem Use-Case-Diagramm lassen sich nun folgende funktionale Anforderungen ableiten:

ID Funktionale Anforderung

F1	Das System ermöglicht die Suche nach Filmen (Titel) und CDs (Titel/Interpret) und zeigt die Ergebnisse in einer Tabellenansicht an.
F2	Das System ermöglicht das Bearbeiten und Löschen bestehender Einträge aus der Tabellenansicht.
F3	Das System öffnet bei erfolgloser Suche automatisch eine Eingabemaske zum Anlegen eines neuen Mediums und speichert dieses nach Bestätigung ab.

Tabelle 1

3. Nichtfunktionale Anforderungen

Neben den grundlegenden Funktionen gibt es weitere Anforderungen an das Projekt. Diese sind in der folgenden Tabelle zusammengefasst:

ID	Nicht-funktionale Anforderung	Messbarkeit / Kriterium
NF1	Die wichtigsten Funktionen (Suche, Anlegen, Bearbeiten, Löschen) müssen leicht erreichbar sein.	Maximal 2 Klicks bis zur Funktion.
NF2	Die Benutzeroberfläche muss klar strukturiert sein und Eingabebereich, Ergebnisbereich und Aktionen eindeutig trennen.	Sichtbare und logisch getrennte Bereiche in jeder GUI-Maske.
NF3	Das System darf im normalen Betrieb nicht abstürzen und alle CRUD-Operationen fehlerfrei ausführen.	Testlauf über mind. 30 Minuten ohne Absturz; 100 % erfolgreiche CRUD-Tests.
NF4	Suchanfragen und das Laden der Tabelle sollen schnell ablaufen.	Antwortzeit ≤ 500 ms pro Suchvorgang.
NF5	Die Anwendung soll vollständig lokal und ohne Internetverbindung nutzbar sein.	Alle Funktionen auch im Flugmodus bzw. ohne Netzwerkanbindung verfügbar.
NF6	Die Anwendung soll ohne Installationsprozess startbar sein.	Start per .bat
NF7	Änderungen sollen nachvollziehbar sein und ohne Build-Fehler möglich sein.	Commit-Historie vollständig; lokaler Build ohne Fehler jederzeit möglich.

Tabelle 2

4. Glossar

Begriff	Definition
CRUD	Create, Read, Update, Delete (Grundoperationen einer Datenbank)
GUI	Graphical User Interface (grafische Benutzeroberfläche)
SQLite	Lokale relationale Datenbank
JavaFX	Framework für grafische Oberflächen in Java
Maven	Build-Tool zur Verwaltung von Abhängigkeiten
GitHub	Plattform zur Versionsverwaltung und Quellcode-Veröffentlichung

Tabelle 3

Abbildungsverzeichnis Anforderungsdokument

Abbildung 1	2
--------------------	----------

Tabellenverzeichnis Anforderungsdokument

Tabelle 1	2
Tabelle 2	3
Tabelle 3	3