

Architekturdokument

1. Technologieübersicht

Der Anwendung Medianizer wird mit modernen Java-Technologien entwickelt, um eine stabile, plattformunabhängige und wartbare Desktopanwendung zu gewährleisten. Zu Beginn der Entwicklung wird zunächst mit Swing gearbeitet, um die grundlegende Funktionalität der Anwendung zu realisieren. Im weiteren Verlauf wird jedoch auf JavaFX umgestellt, da dieses Framework eine moderne und flexiblere Benutzeroberfläche ermöglicht. Die Umstellung erfolgt also im Zuge der Weiterentwicklung des Projekts. Es folgt eine Tabelle mit den verwendeten Technologien bzw. Werkzeugen:

Komponente	Technologie/Werkzeug	Begründung
Programmiersprache	Java 21	Moderne, objektorientierte Sprache mit stabiler JVM und langfristigem Support
GUI-Framework	JavaFX 21	Grafische Benutzeroberflächen
Datenbank	SQLite	Leichte, dateibasierte SQL-Datenbank ohne Serverinstallation
Build-Management	Apache Maven	Automatisiertes Build-System mit Abhängigkeitsverwaltung und Plugins
IDE	Eclipse	Entwicklungsumgebung mit Maven-Integration und JavaFX-Unterstützung.
Versionsverwaltung	Git / GitHub	Versionskontrolle
Bibliotheken	org.openjfx, org.xerial	Für JavaFX-GUI und SQLite-Datenbankanbindung.

Tabelle 1

2. Architekturübersicht

Die Anwendung folgt einer klassischen Drei-Schichten-Architektur (Model-View-Controller), die eine klare Trennung von Datenhaltung, Anwendungslogik und Darstellung ermöglicht:

1. Model-Schicht – enthält Datenklassen und den DatabaseManager für CRUD-Operationen in SQLite

2. Controller-Schicht – enthält Logik zur Steuerung der Interaktion zwischen Benutzeroberfläche und Datenbank
3. View-Schicht – umfasst die JavaFX-Komponenten zur Darstellung der Benutzeroberfläche

Diese Architektur erleichtert Wartung, Erweiterung und Wiederverwendbarkeit der Komponenten.

3. Struktur

In der Model-Schicht befinden sich die Klassen Media, Film/CD und DatabaseManager. Media ist hierbei die Basisklasse für alle Medientypen und hat die Attribute ID, Titel, Anzahl und Typ. Film und CD sind ihre Spezialisierungen. Film enthält hier noch keine weiteren Attribute, ist aber dennoch als Spezialisierung festgelegt, um sie leicht erweiterbar zu gestalten. CD enthält das zusätzliche Attribut Interpret. Der DatabaseManager verwaltet die Datenbankzugriffe (CRUD).

Die Controller-Schicht enthält insgesamt vier Listener-Klassen (MediaListenerFX, InsertButtonListenerFX, ThirdGUIListenerFX und OkButtonListener), die Benutzerinteraktionen verarbeiten und das Datenmodell und die GUI aktualisieren.

Die View-Schicht enthält die Klassen zur Darstellung der GUI (FirstGUIFX, SecondGUIFX, ThirdGUI und SelectPanelFX). Das SelectPanelFX wird dabei sowohl in der FirstGUIFX, als auch in der SecondGUIFX verwendet. Dies ist auch der Grund es als eigene Klasse zu definieren. Außerdem sind in dieser Schicht auch die Interfaces InputProvider, InsertProvider und TypeSelectionCallback enthalten, welche in den Listenern dazu dienen die Eingaben aus der jeweiligen GUI zu verarbeiten, ohne GUI-Elemente übergeben zu müssen, da dies der MVC-Struktur widerspräche.

Zusätzlich wurden oberhalb des MVC-Modells die beiden Klassen Constants und MainApp erstellt. MainApp startet hierbei die Anwendung und Constants enthält Konstanten, die in verschiedenen Bereichen des Projekts Verwendung finden und daher global angelegt wurden.

Es folgen UML-Klassendiagramme für die Architekturelemente Model, View und Controller:

Model

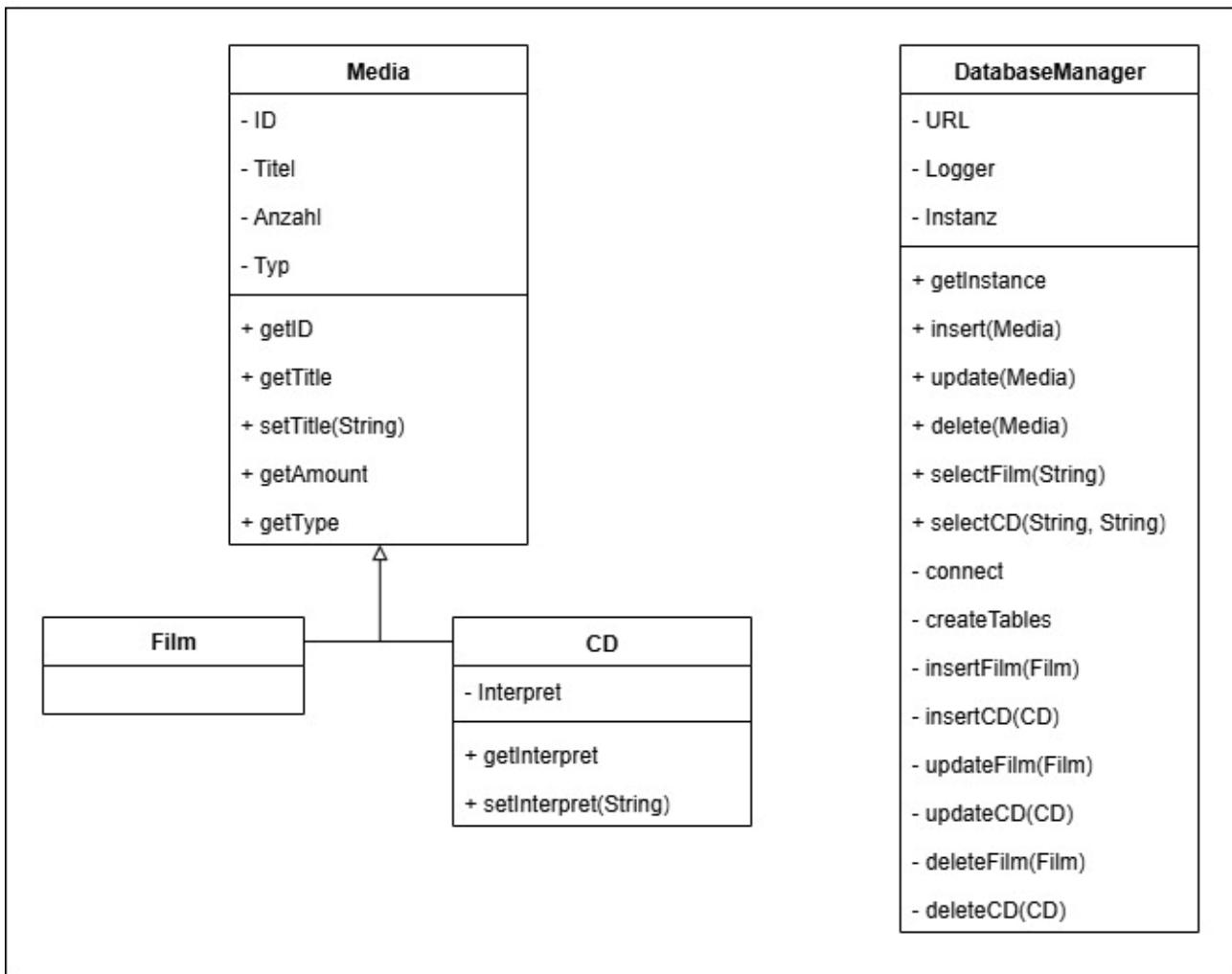


Abbildung 1

View

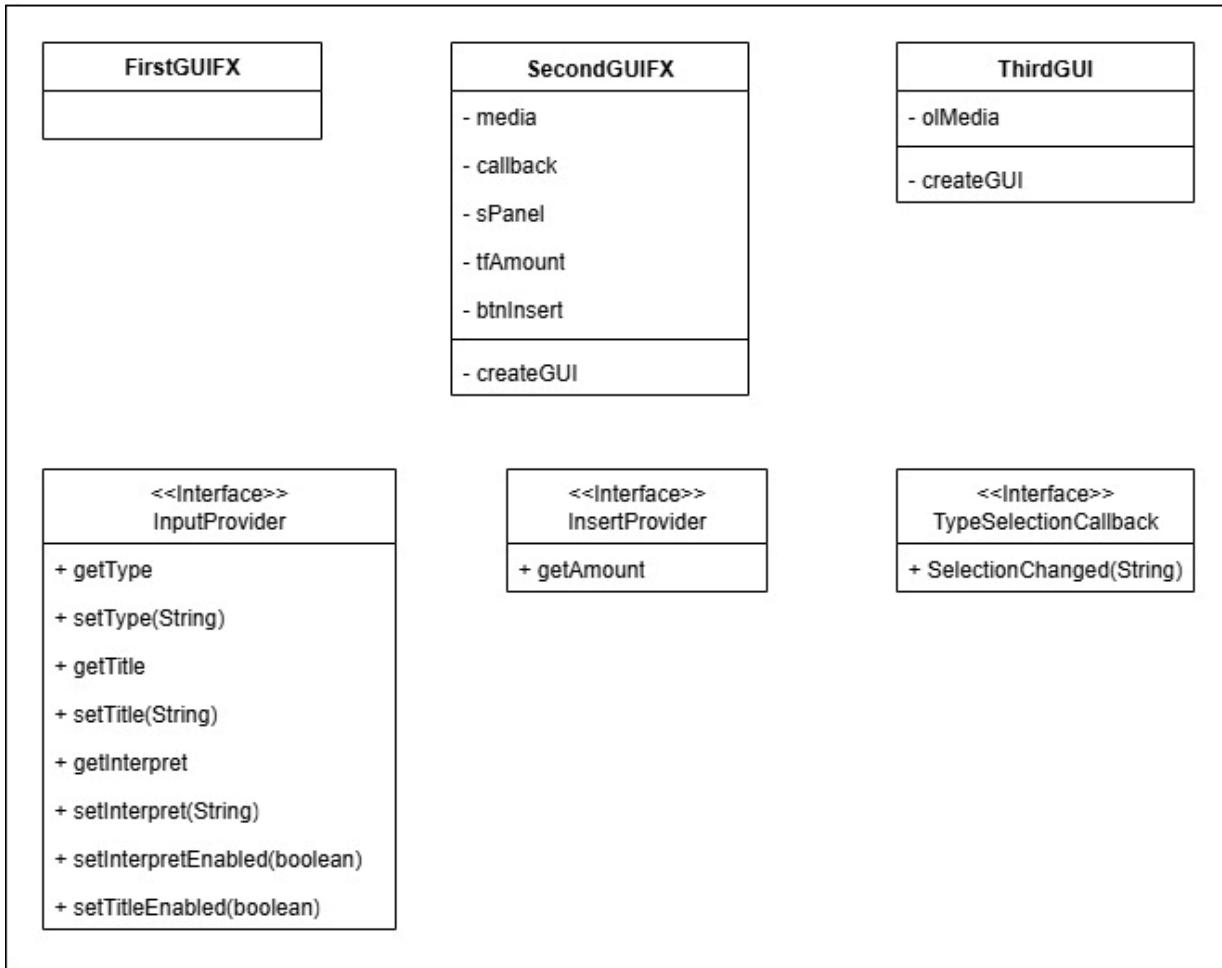


Abbildung 2

Controller

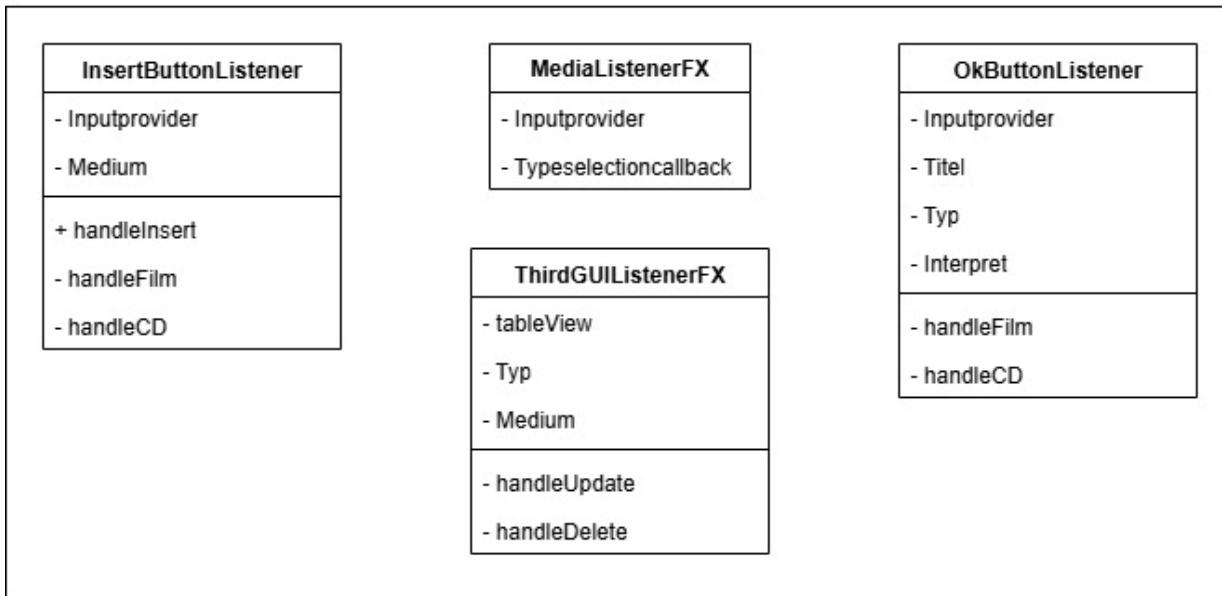


Abbildung 3

4. Verhalten

Ein zentraler Systemablauf ist das Suchen in der Benutzerschnittstelle eins. Für diesen Ablauf wird von einem Erfolg ausgegangen. Dementsprechend wird die Benutzerschnittstelle drei mit den gefundenen Datensätzen aufgerufen. Dieser wird im Folgenden als Sequenzdiagramm dargestellt:

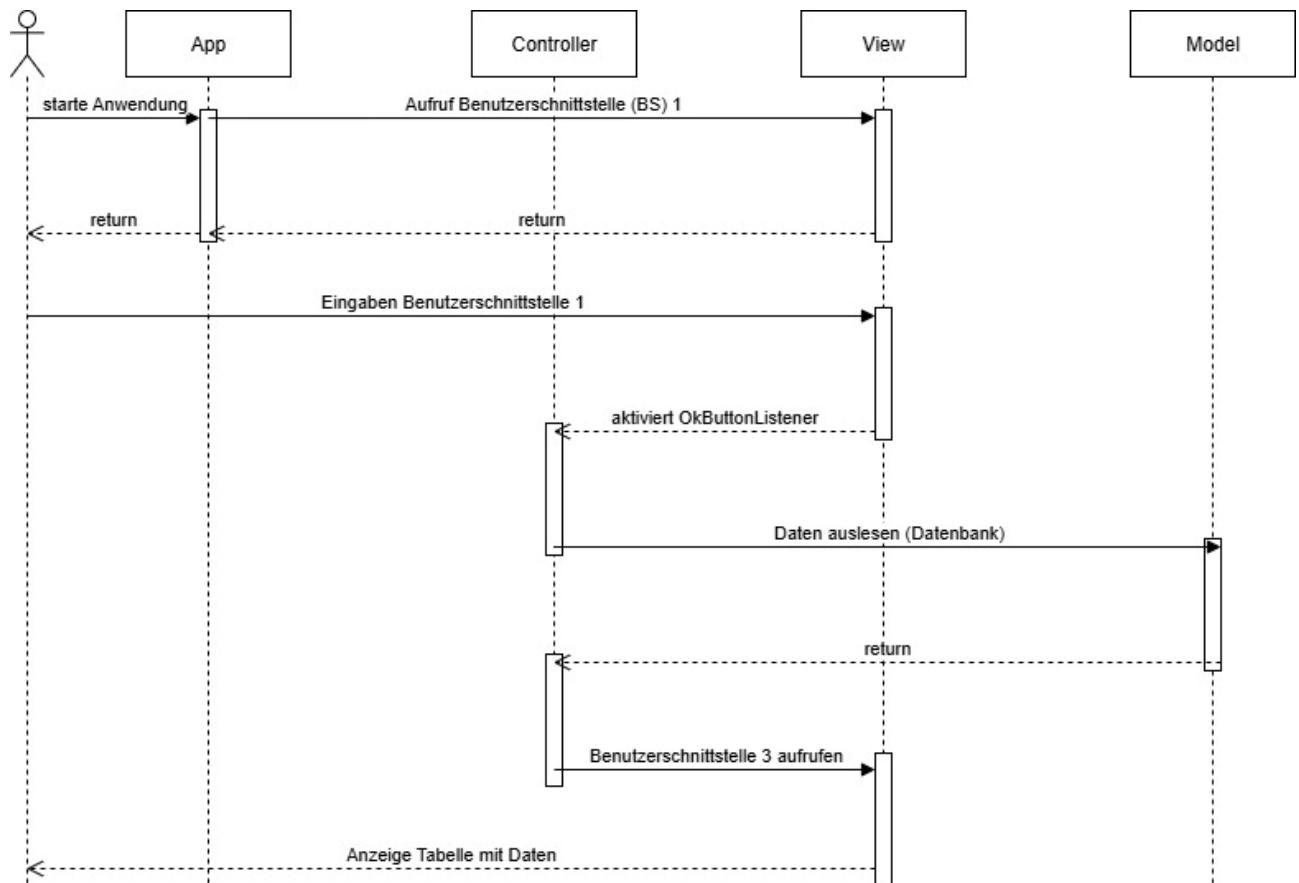


Abbildung 4

Abbildungsverzeichnis Architekturdokument

Abbildung 1	3
Abbildung 2	4
Abbildung 3	4
Abbildung 4	5

Tabellenverzeichnis Architekturdokument

Tabelle 1	1
------------------	----------