

**Hausarbeit**im Bildungsgang

„Staatlich geprüfte/r Wirtschaftsinformatiker/in“

gemäß §5 der Ausbildungs- und Prüfungsordnung

Entwicklung einer lokalen Media-Player-Anwendung zur Verwaltung und Wiedergabe von Audio- und Videodateien

vorgelegt von: Haris Dervovic

Klasse: HBFS – WI23ZA2

Adresse: Ebersteinstraße 2

Ort: 66117 Saarbrücken

E-Mail: [harisdervovic2003@gmail.com](mailto:harisdervovic2003@gmail.com)

Abgabetermin: 15.05.2025

Betreuer/in: Herr Schuler

**Inhaltsverzeichnis**

[1. Pflichtenheft/Lastenheft: Mediaplayer 4](#_Toc197800672)

[1.1. Einführung 4](#_Toc197800673)

[1.2. Ist-Situation 4](#_Toc197800674)

[1.3. Soll-Situation 5](#_Toc197800675)

[1.3.1 Soll-Zustand 5](#_Toc197800676)

[1.3.2 Funktionale Anforderungen 5](#_Toc197800677)

[1.3.3 Nicht-Funktionale Anforderungen 6](#_Toc197800678)

[1.3.4 Schnittstellen 6](#_Toc197800679)

[1.3.5 Risiken 6](#_Toc197800680)

[1.4. Abnahmekriterien 6](#_Toc197800681)

[1.4.1 Muss-Kriterien 6](#_Toc197800682)

[1.4.2 Kann-Kriterien 7](#_Toc197800683)

[1.5. Use-Case-Diagramm 7](#_Toc197800684)

[1.5.1 Detailbeschreibung der zentralen Use-Cases 8](#_Toc197800685)

[1.6. Projektplan 9](#_Toc197800686)

[1.7. Produktumgebung 9](#_Toc197800687)

[1.8. Skizze von GUI 10](#_Toc197800688)

[1.9. DB-Entwurf 13](#_Toc197800689)

[1.10. Link zu einem gehosteten Git-Repository 14](#_Toc197800690)

[1.11. Testplan 14](#_Toc197800691)

[2. Benutzerhandbuch 16](#_Toc197800692)

[2.1 Start der Anwendung 16](#_Toc197800693)

[2.2 Login und Registrierung 17](#_Toc197800694)

[2.3 Startseite und die Navigation in der Anwendung 19](#_Toc197800695)

[2.4 Mediathek 20](#_Toc197800696)

[2.5 Datei der Mediathek hinzufügen 21](#_Toc197800697)

[2.6 Playlists 22](#_Toc197800698)

[2.7 Medienwiedergabe und Steuerung 23](#_Toc197800699)

[2.8 YouTube Downloader 25](#_Toc197800700)

[3. Implementierung 28](#_Toc197800701)

[3.1 Überblick über die Projektstruktur 28](#_Toc197800702)

[3.2 Klassenstruktur und Datenmodell 28](#_Toc197800703)

[3.2.1 Medium 29](#_Toc197800704)

[3.2.2 Playlist 29](#_Toc197800705)

[3.2.3 PlaylistMedium 29](#_Toc197800706)

[3.2.4 Klassendiagramm 30](#_Toc197800707)

# 1. Pflichtenheft/Lastenheft: Mediaplayer

## 1.1. Einführung

Max Mustermann, Geschäftsführer des Medienunternehmens "VisionMedia", benötigt eine Software, die den Arbeitsalltag seiner Mitarbeiter erleichtert. Das Unternehmen erstellt regelmäßig Präsentationen, Podcasts und Social-Media-Inhalte und möchte daher eine zentrale Plattform zur Organisation und Wiedergabe lokal gespeicherter Medien. Zusätzlich besteht Interesse an einer optionalen Funktion, die es ermöglicht, Inhalte von Plattformen wie YouTube herunterzuladen und direkt in die Mediathek zu integrieren. Während diese Funktion aktuell als potenzielle Erweiterung geplant ist, liegt der Fokus zunächst auf einer benutzerfreundlichen Software für lokale Medienverwaltung und -wiedergabe. Das Projekt simuliert ein realitätsnahes Szenario, in dem Software flexibel weiterentwickelt werden kann, um künftige Anforderungen zu erfüllen.

## 1.2. Ist-Situation

Das Medienunternehmen VisionMedia nutzt derzeit verschiedene Programme, um lokale Medieninhalte zu verwalten und abzuspielen. Diese Programme sind oft unübersichtlich, bieten keine einheitliche Benutzeroberfläche und sind nicht optimal auf die Bedürfnisse des Unternehmens zugeschnitten. Darüber hinaus wäre es für die Mitarbeiter hilfreich, externe Inhalte von Plattformen wie YouTube einfach in ihre Mediathek integrieren zu können, um diese in Präsentationen oder Social-Media-Projekten zu verwenden. Aktuell müssen dafür zusätzliche Tools genutzt werden, was den Arbeitsprozess unnötig verkompliziert. Eine zentrale Lösung, die lokale Medienverwaltung und optional die Integration externer Inhalte vereint, würde die Effizienz und den Workflow des Unternehmens deutlich verbessern.

## 1.3. Soll-Situation

### 1.3.1 Soll-Zustand

Der Mediaplayer MediaSphere bietet eine benutzerfreundliche Oberfläche, um Medieninhalte (MP3 und MP4) lokal abzuspielen, zu organisieren und in Playlists zu verwalten. Der Kundenwunsch wird erfüllt, indem Zeit und Aufwand bei der Mediennutzung reduziert und eine zentrale Plattform zur Verwaltung geschaffen wird.

### 1.3.2 Funktionale Anforderungen

1. **Import von Mediendateien in die Bibliothek** (15 Stunden)
   * Funktion, um Mediendateien durch Dateiimport hinzuzufügen, Dies ermöglicht die Nutzung der Medienbibliothek und aller anderen Funktionen.
2. **Abspielen von MP3-Dateien** (10 Stunden)
   * Funktion, um MP3-Dateien zu laden und wiederzugeben.
3. **Abspielen von MP4-Dateien** (12 Stunden)
   * Funktion, um MP4-Videodateien zu laden und abzuspielen.
4. **Playlist-Verwaltung** (20 Stunden)
   * Erstellen, Bearbeiten und Speichern von Wiedergabelisten.
5. **Suchfunktion** (15 Stunden)
   * Möglichkeit, nach Titeln oder Kategorien zu suchen.
6. **Grafische Benutzeroberfläche (GUI)** (20 Stunden)
   * Die Software soll über eine grafische Benutzeroberfläche bedienbar sein.
7. **Login-System** (30 Stunden)
   * Ein System, bei dem Nutzer sich registrieren und anmelden können, um eigene Playlists zu verwalten.
   * Ohne Anmeldung können Standardfunktionen wie das Abspielen von Dateien genutzt werden, aber keine Playlist-Funktionalitäten.
8. **Standardfunktionen eines Mediaplayers** (10 Stunden)
   * Lautstärkeregler, Loop-Funktion und Skip-Funktion für Titel.

### 1.3.3 Nicht-Funktionale Anforderungen

1. **GUI-Design** (10 Stunden)
   * Die grafische Benutzeroberfläche soll gut und modern aussehen.
2. **Stabilität** (12 Stunden)
   * Absturzsicher und zuverlässige Funktionalität.

### 1.3.4 Schnittstellen

Der Mediaplayer verarbeitet in erster Linie lokal gespeicherte Dateien und benötigt hierfür keine externen Schnittstellen. Sollte die optionale Funktion eines YouTube-Downloaders aktiviert werden, erfolgt ein Zugriff auf die YouTube-Plattform über die yt-dlp-Bibliothek.

### 1.3.5 Risiken

1. **Zeitmangel:** Begrenzte Zeit für die Implementierung und Testphase.
2. **Technische Komplexität:** Schwierigkeiten bei der Integration verschiedener Funktionalitäten (z. B. Playlist-Management).
3. **Unvorhergesehene Bugs:** Unerwartete Fehler könnten die Entwicklungszeit verlängern.

## 1.4. Abnahmekriterien

Die Abnahmekriterien sind in Muss-Kriterien und Kann-Kriterien unterteilt.

### 1.4.1 Muss-Kriterien

1. Die Software ermöglicht den Import von Mediendateien in die Bibliothek.
2. Die Software bietet ein Login-System, das Nutzern ermöglicht, sich zu registrieren und anzumelden, um eigene Playlists zu verwalten.
3. Die Software kann MP3- und MP4-Dateien fehlerfrei abspielen.
4. Benutzer können Playlists erstellen, bearbeiten und speichern.
5. Die Suchfunktion liefert korrekte Ergebnisse.
6. Die Benutzeroberfläche ist ansprechend und barrierefrei.
7. Stabilität: Die Software läuft ohne Abstürze.
8. Lautstärkeregler, Loop-Funktion und Skip-Funktion sind implementiert.

### 1.4.2 Kann-Kriterien

1. Unterstützung zusätzlicher Audio- und Videoformate wie WAV oder AVI.
2. Erweiterte Suchfilter (z. B. nach Album oder Genre).
3. Anpassbare Benutzeroberfläche (z. B. Farbschema ändern, Darkmode).
4. **YouTube-Downloader:** Integration eines Downloaders für Videos und Audios über yt-dlp.

## 1.5. Use-Case-Diagramm

Ein Bild, das Diagramm, Text, Reihe, Kreis enthält.

Automatisch generierte Beschreibung

### 1.5.1 Detailbeschreibung der zentralen Use-Cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case | Akteur(e) | Beschreibung | Vorbedingungen | Nachbedingungen |
| Registrieren/Anmelden | Benutzer | Der Benutzer registriert sich oder meldet sich mit bestehenden Daten an. | Keine Registrierung vorhanden oder abgemeldet. | Benutzer ist angemeldet und hat Zugriff auf Funktionen |
| Medien importieren | Benutzer | Der Benutzer fügt Mediendateien (MP3, MP4) zur Bibliothek hinzu. | Benutzer ist eingeloggt. | Mediendateien sind gespeichert und verfügbar. |
| Medien durchsuchen | Benutzer | Der Benutzer durchsucht die Bibliothek nach Titeln oder Interpreten. | Mediendateien müssen vorhanden sein. | Gefundene Medien werden angezeigt. |
| Medien abspielen | Benutzer | Der Benutzer spielt eine Audio- oder Videodatei aus der Bibliothek oder einer Playlist ab. | Benutzer ist eingeloggt und Medien sind importiert. | Medium läuft und Benutzer kann es steuern. |
| Playlist verwalten | Benutzer | Der Benutzer erstellt, bearbeitet oder speichert Playlists, um Medien zu gruppieren. | Benutzer ist eingeloggt. | Playlist wird gespeichert und kann verwendet werden. |
| Mediensteuerung | Benutzer | Der Benutzer steuert die Wiedergabe: Lautstärke ändern, Skippen, Loop aktivieren. | Wiedergabe ist gestartet. | Wiedergabe wird entsprechend gesteuert. |

## 1.6. Projektplan

Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

Automatisch generierte Beschreibung

## 1.7. Produktumgebung

* **Technologien:** C#, WPF, .NET , Visual Studio, yt-dlp, eventuell LibVLC. Eventuell eine externe UI-Framework-Bibliothek (wie z.B. MaterialDesignInXAML oder MahApps.Metro)
* **Datenbank:** SQLite wird verwendet, um eine leichtgewichtige und lokal nutzbare Datenbanklösung bereitzustellen.

## 1.8. Skizze von GUI

Ein Bild, das Screenshot, Text, Diagramm, Rechteck enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Diagramm, Rechteck enthält.

Automatisch generierte Beschreibung

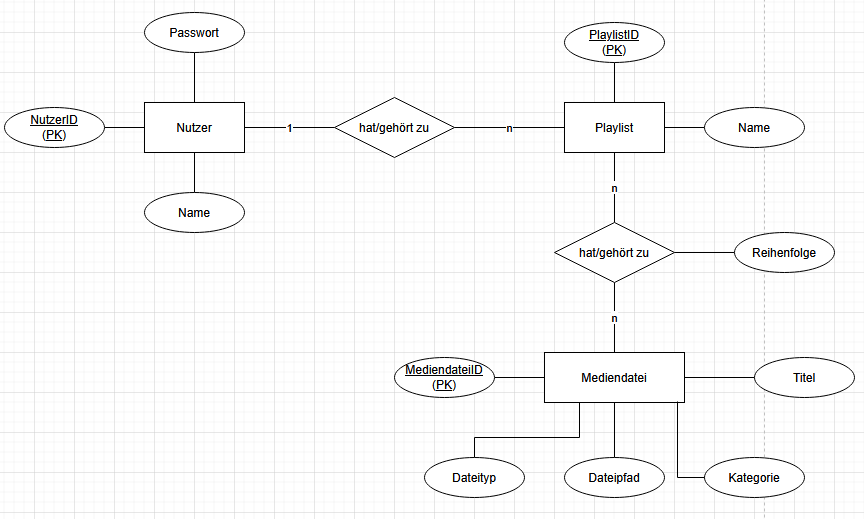
Ein Bild, das Text, Screenshot, Zahl, parallel enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Zahl, parallel enthält.

Automatisch generierte Beschreibung

## 1.9. DB-Entwurf



## 1.10. Link zu einem gehosteten Git-Repository

https://github.com/HarisDervovic/MediaSphere.git

## 1.11. Testplan

**Testfälle für Muss-Kriterien**

**ID:** T01  
**Beschreibung:** Die Software kann MP3- und MP4-Dateien fehlerfrei abspielen.  
**Vorbedingungen:** Die Anwendung ist gestartet, und eine Mediendatei ist verfügbar.  
**Test-Schritte:**

1. Mediathek öffnen.
2. Auswahl einer MP3- oder MP4-Datei.
3. Klicken auf den "Abspielen"-Button.  
   **Erwartetes Resultat** Die Wiedergabe der Datei beginnt ohne Verzögerung.

**ID:** T02  
**Beschreibung:** Benutzer können Playlists erstellen, bearbeiten und speichern.  
**Vorbedingungen:** Die Anwendung ist gestartet, und mindestens eine Mediendatei ist verfügbar.  
**Test-Schritte:**

1. Wechsel in den Playlist-Manager.
2. Erstellen einer neuen Playlist.
3. Hinzufügen einer Mediendatei zur Playlist.
4. Speichern der Playlist unter einem benutzerdefinierten Namen.

**Erwartetes Resultat:** Die Playlist wird gespeichert und ist erneut abrufbar.

**ID:** T03  
**Beschreibung:** Die Suchfunktion liefert korrekte Ergebnisse.  
**Vorbedingungen:** Die Anwendung ist gestartet, und die Medienbibliothek ist mit mehreren Dateien gefüllt.  
**Test-Schritte:**

1. Eingabe eines Suchbegriffs in die Suchleiste innerhalb der Mediathek.
2. Klicken auf den "Suchen"-Button.  
   **Erwartetes Resultat:** Passende Titel werden in den Suchergebnissen angezeigt.

**ID:** T04  
**Beschreibung:** Die Benutzeroberfläche ist ansprechend und barrierefrei.  
**Vorbedingungen:** Die Anwendung ist gestartet.  
**Test-Schritte:**

1. Starten der Anwendung.
2. Navigieren durch die Benutzeroberfläche.
3. Überprüfen auf klare Bedienbarkeit und Zugänglichkeit.  
   **Erwartetes Resultat:** Die Oberfläche ist visuell ansprechend, und alle Funktionen sind einfach zugänglich.

**ID:** T05  
**Beschreibung:** Stabilität: Die Software läuft ohne Abstürze.  
**Vorbedingungen:** Die Anwendung ist gestartet und wird über längere Zeit genutzt.  
**Test-Schritte:**

1. Zufällige Funktionen der Anwendung für mehrere Stunden nutzen.  
   **Erwartetes Resultat:** Keine Abstürze oder Fehlermeldungen.

**ID:** T06  
**Beschreibung:** Lautstärkeregler, Loop-Funktion und Skip-Funktion sind implementiert.  
**Vorbedingungen:** Die Anwendung spielt eine Datei ab.  
**Test-Schritte:**

1. Ändern der Lautstärke über den Regler.
2. Aktivieren der Loop-Funktion.
3. Überspringen eines Titels mithilfe der Skip-Taste.  
   **Erwartetes Resultat:** Alle Funktionen reagieren wie erwartet.

**Testfälle für Kann-Kriterien**

**ID:** K01  
**Beschreibung:** Integration eines YouTube-Downloaders.  
**Vorbedingungen:** yt-dlp ist in der Anwendung integriert.  
**Test-Schritte:**

1. Eingeben eines YouTube-Links in das Downloader-Feld.
2. Klicken auf "Download".  
   **Erwartetes Resultat:** Die Datei wird heruntergeladen und der Bibliothek hinzugefügt.

# 2. Benutzerhandbuch

## 2.1 Start der Anwendung

Die Anwendung MediaSphere wird über die Datei **MediaSphere.exe** gestartet.  
Nach dem Start wird standardmäßig die Login-Seite angezeigt. Dort kann sich ein Benutzer anmelden, registrieren oder als Gast fortfahren.

Wenn bei einem vorherigen Login die Option „Angemeldet bleiben“ aktiviert wurde, wird der gespeicherte Benutzername automatisch geladen. In diesem Fall wird die Login-Seite übersprungen und der Benutzer direkt zur Hauptanwendung weitergeleitet.

Wird MediaSphere durch Öffnen einer Mediendatei (zum Beispiel durch Doppelklick im Windows Explorer) gestartet, prüft das Programm ebenfalls den Anmeldestatus:

* Ist „Angemeldet bleiben“ aktiv, wird die Datei direkt mit dem gespeicherten Benutzer geöffnet.
* Ist kein Benutzer gespeichert, wird automatisch der Gast-Modus aktiviert. Auch hier startet die Wiedergabe, jedoch ohne Zugriff auf persönliche Playlists.

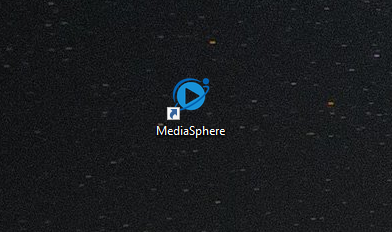


Abbildung 1: Verknüpfung zur MediaSphere-Anwendung auf dem Desktop

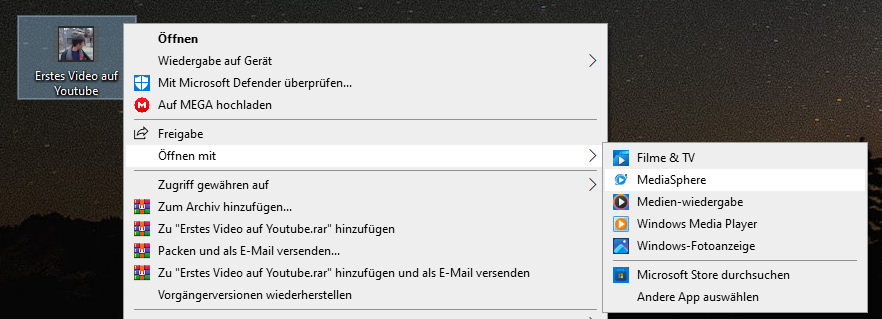


Abbildung 2: Öffnen einer Videodatei über das Kontextmenü im Windows Explorer mit MediaSphere

## 2.2 Login und Registrierung

Nach dem Start der Anwendung erscheint die Login-Seite.  
Hier kann sich der Benutzer mit einem bestehenden Benutzernamen und Passwort anmelden. Die Anmeldedaten werden verschlüsselt und mit der Datenbank verglichen. Bei erfolgreicher Anmeldung wird der Benutzer zur Hauptanwendung weitergeleitet.

Ist die Option „Angemeldet bleiben“ aktiviert, wird der Benutzername lokal gespeichert. Beim nächsten Start wird dann automatisch eingeloggt – die Login-Seite wird in diesem Fall übersprungen.

Alternativ kann über den Button „Als Gast fortfahren“ auch ohne Anmeldung auf die Anwendung zugegriffen werden. In diesem Modus sind grundlegende Funktionen wie das Abspielen von Mediendateien verfügbar, personalisierte Funktionen wie Playlists jedoch deaktiviert.

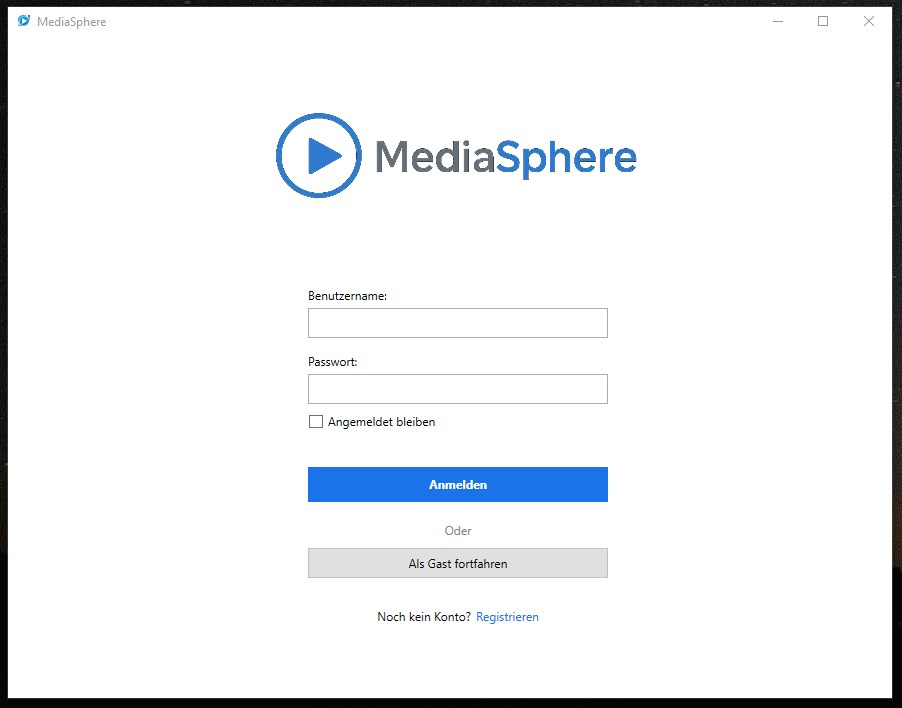


Abbildung 3: Login-Seite von MediaSphere mit Benutzername, Passwortfeld, „Angemeldet bleiben“, Gast-Modus und Registrieren-Link

Wenn der Benutzer noch kein Konto hat, kann über den Link „Registrieren“ eine neue Benutzerregistrierung gestartet werden.

Auf der Registrierungsseite gibt der Benutzer einen Benutzernamen, ein Passwort und eine Passwortbestätigung ein. Die Passwörter müssen exakt übereinstimmen – andernfalls erscheint ein Warnhinweis, und der Registrieren-Button bleibt deaktiviert. Erst bei gültigen Eingaben wird die Schaltfläche aktiviert.

Nach Klick auf „Registrieren“ wird geprüft, ob der Benutzername bereits existiert. Ist das nicht der Fall, wird der neue Benutzer mit verschlüsseltem Passwort in der Datenbank gespeichert. Anschließend erscheint eine Erfolgsmeldung, und der Benutzer wird zur Login-Seite zurückgeleitet. Falls der Benutzername bereits vergeben ist, wird eine entsprechende Fehlermeldung angezeigt.

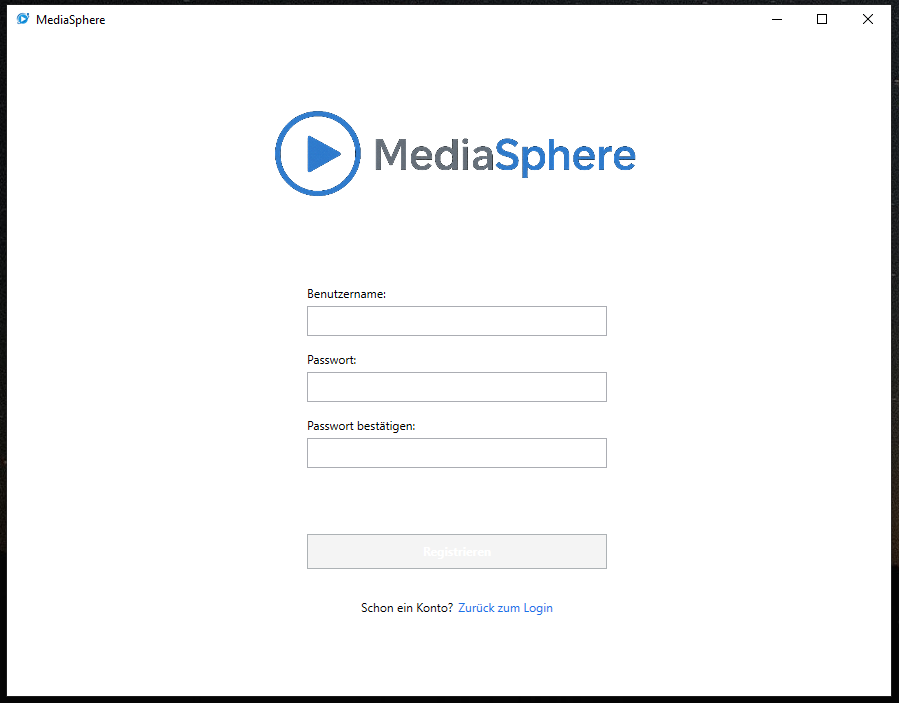


Abbildung 4: Registrierungs-Seite mit Eingabefeldern, Passwort-Bestätigung, Fehlermeldung bei Nicht-Übereinstimmung und Button „Zurück zum Login“

## 2.3 Startseite und die Navigation in der Anwendung

Nach dem erfolgreichen Login oder dem Start im Gast-Modus wird die Startseite von MediaSphere angezeigt. Im rechten Bereich wird eine Begrüßung eingeblendet mit dem Hinweis, dass die Navigation über die linke Seitenleiste erfolgt.

Die linke Sidebar ist dauerhaft sichtbar und enthält alle Hauptfunktionen der Anwendung. Sie ist übersichtlich gestaltet, farblich an das Farbschema von MediaSphere angepasst und reagiert auf Mausbewegung sowie aktive Auswahl.

Folgende Funktionen stehen über die Sidebar zur Verfügung:

* **Startseite:** Zurück zur Willkommensansicht
* **Datei der Mediathek hinzufügen:** Ermöglicht das Hinzufügen neuer Audio- oder Videodateien
* **Mediathek:** Zeigt die gesamte Medienbibliothek an, inklusive Such- und Filterfunktionen
* **Playlists:** Ermöglicht das Erstellen, Bearbeiten und Abspielen eigener Playlists (nur verfügbar, wenn kein Gast-Modus aktiv ist)
* **YouTube Downloader:** Bietet die Möglichkeit, Videos über yt-dlp herunterzuladen und in die Mediathek zu integrieren
* **Abmelden:** Beendet die aktuelle Sitzung und startet die Anwendung neu, um einen neuen Benutzer anzumelden

Wenn der Benutzer als Gast angemeldet ist, ist die Playlist-Schaltfläche deaktiviert, da diese Funktion nur für registrierte Benutzer verfügbar ist.

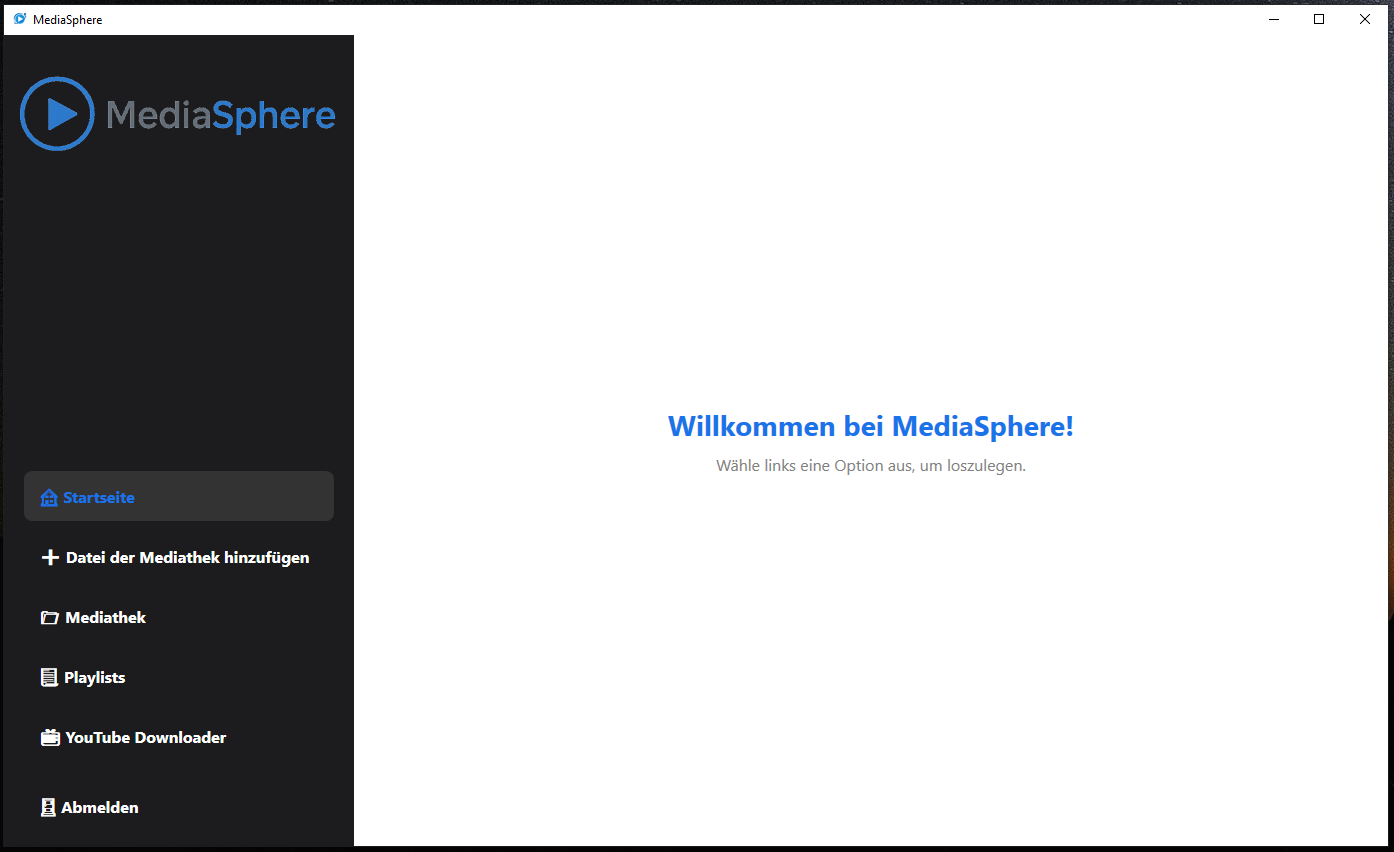


Abbildung 5: Startseite und linke Navigation in MediaSphere

## 2.4 Mediathek

Die Mediathek zeigt alle importierten Mediendateien (Audio und Video) in einer übersichtlichen Liste. Jeder Eintrag enthält Informationen wie Titel, Typ (z. B. mp3 oder mp4) und Kategorie. Die Ansicht ist optisch an das MediaSphere-Design angepasst.

Im oberen Bereich befindet sich ein **Suchfeld**, mit dem die Liste nach Titel oder Kategorie gefiltert werden kann. Rechts daneben befindet sich ein **Dropdown-Menü**, mit dem die Anzeige auf bestimmte Dateitypen eingeschränkt werden kann (Alle, mp3, mp4).

Jeder Medieneintrag enthält drei Aktionsschaltflächen:

* **▶️ Abspielen**  
  Spielt die Datei direkt über die integrierte Wiedergabeleiste oder das Video-Overlay ab
* **➕ Zur Playlist hinzufügen**  
  Öffnet ein separates Dialogfenster, in dem der Benutzer eine bestehende Playlist auswählen kann. Nach Auswahl und Bestätigung wird das Medium der gewählten Playlist hinzugefügt.
* **🗑️ Löschen**  
  Entfernt die Datei sowohl aus der Datenbank als auch vom Dateisystem

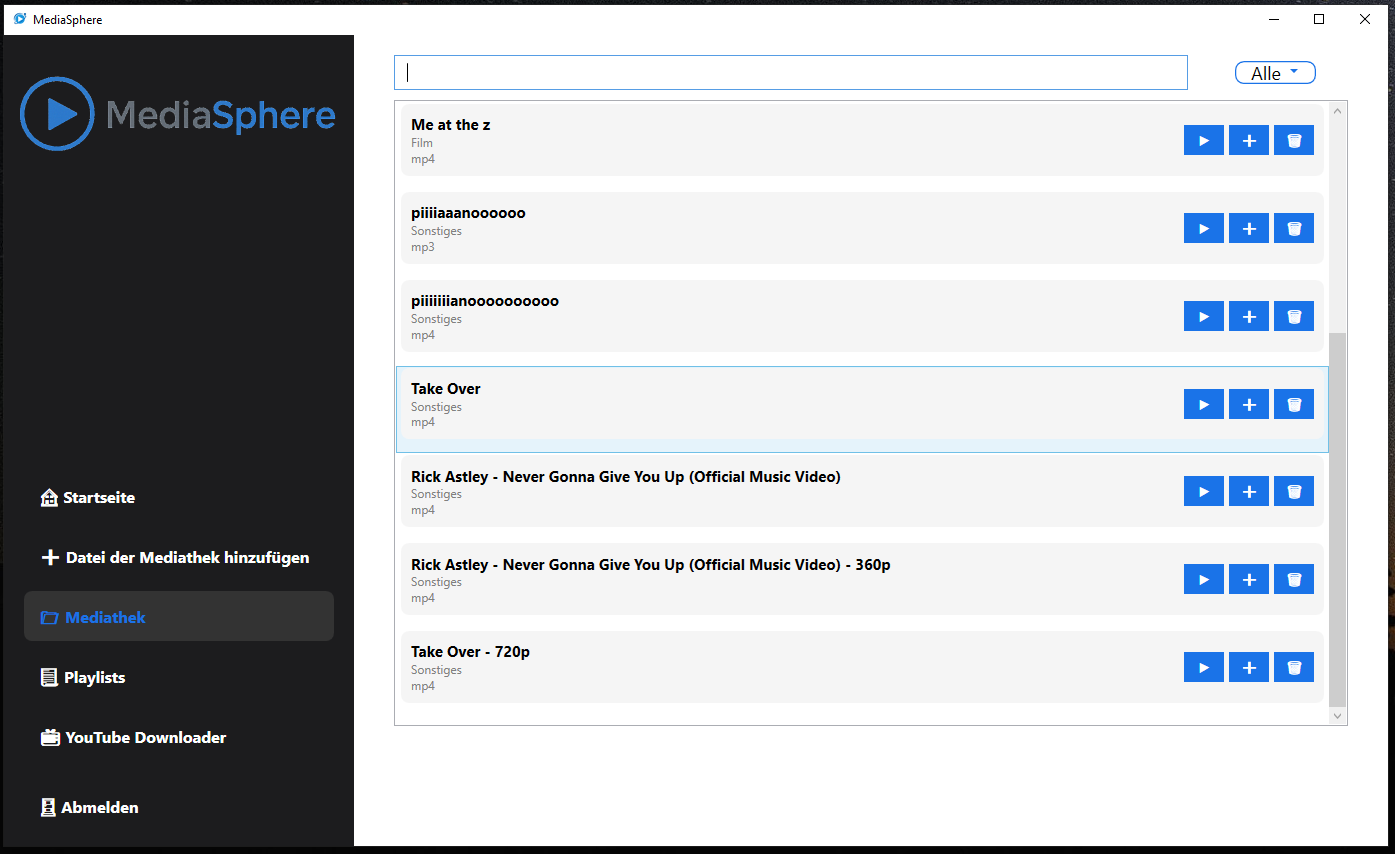


Abbildung 6: Mediathek mit Medienliste, Such- und Filterfunktion sowie Aktionsbuttons

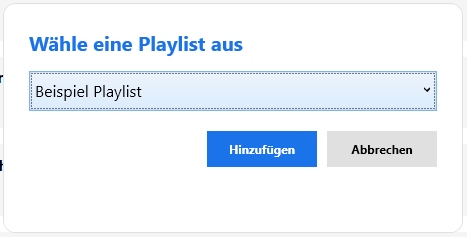


Abbildung 7: Dialogfenster zur Playlist-Zuweisung

## 2.5 Datei der Mediathek hinzufügen

Über die gleichnamige Seite in der Anwendung kann eine neue Mediendatei (z. B. mp3 oder mp4) zur Mediathek hinzugefügt werden.

Dabei wird die Datei physisch in den MediaSphere-Anwendungsordner kopiert und mit ihren Metadaten in der Datenbank gespeichert.

Die Oberfläche der Seite ist klar strukturiert. Die folgenden Eingaben sind erforderlich:

* **Dateipfad:**  
  Über den Ordner-Button kann eine Datei im Explorer ausgewählt werden.  
  Der Pfad wird automatisch eingetragen, der Titel und Typ werden vorausgefüllt.
* **Titel:**  
  Vorgeschlagen auf Basis des Dateinamens, kann aber vom Benutzer angepasst werden.
* **Dateityp:**  
  Wird automatisch aus der Dateiendung (z. B. mp3, mp4) ermittelt und ist schreibgeschützt.
* **Kategorie:**  
  Auswahl aus vordefinierten Kategorien wie Musik, Film, Hörbuch, Podcast, Sonstiges.

Der Button **„➕ Datei der Mediathek hinzufügen“** wird nur aktiviert, wenn alle Felder korrekt ausgefüllt sind.

Nach dem Klick wird:

1. Die Datei in den internen MediaSphere-Mediathek-Ordner kopiert (falls sie dort noch nicht existiert)
2. Der Eintrag mit Pfad, Titel, Typ und Kategorie in der SQLite-Datenbank gespeichert
3. Ein Dialogfenster zur Bestätigung angezeigt

Falls die Datei bereits vorhanden ist oder ein Fehler auftritt, erscheint eine entsprechende Fehlermeldung.

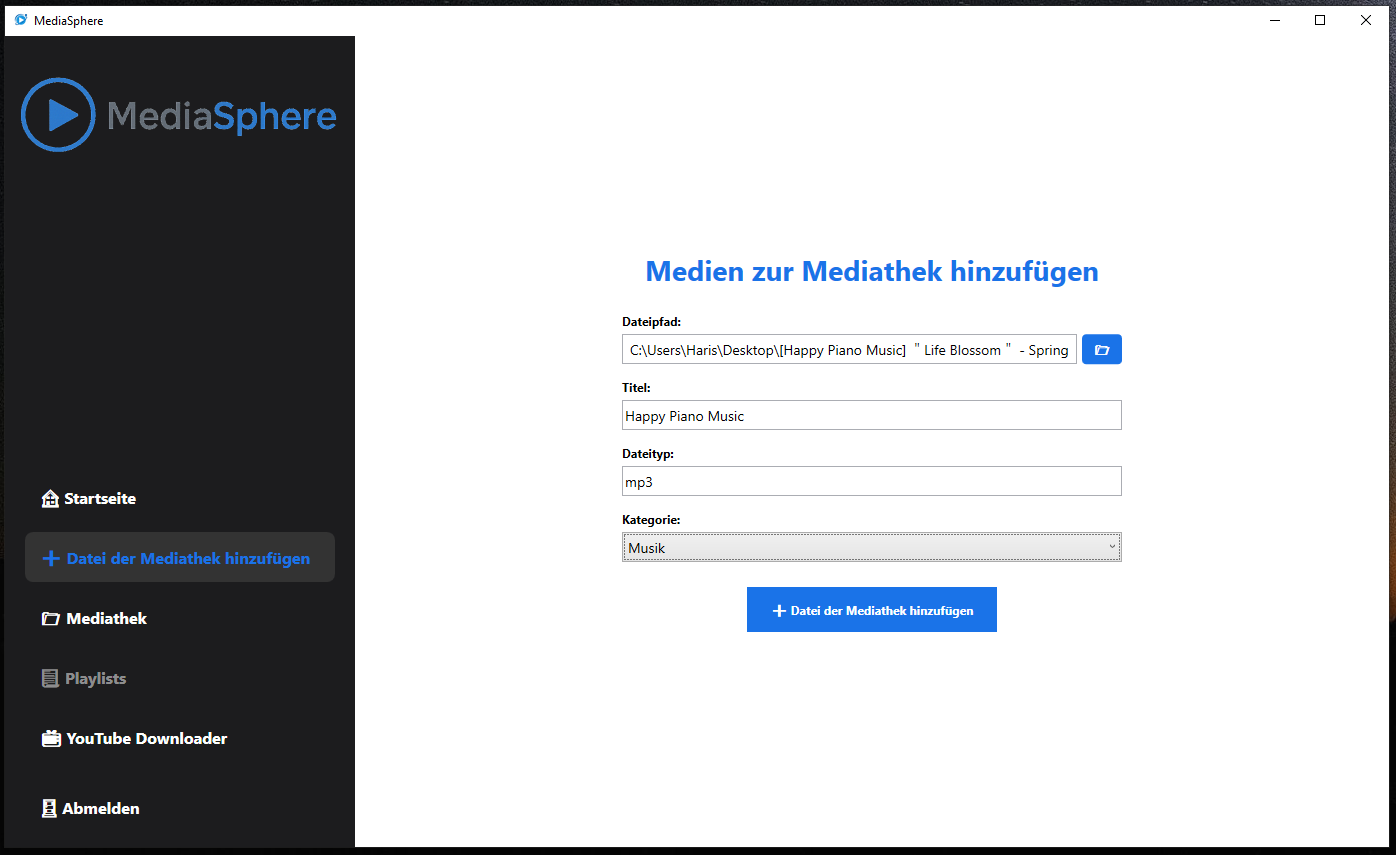


Abbildung 8: Oberfläche zum Hinzufügen einer Datei zur Mediathek

## 2.6 Playlists

In diesem Bereich können Benutzer individuelle Playlists erstellen, anzeigen, abspielen, bearbeiten und löschen. Alle Playlists sind benutzergebunden und werden lokal in der Datenbank gespeichert.

**Neue Playlist erstellen**

Oben auf der Seite kann durch Eingabe eines Namens und Klick auf **„Erstellen“** eine neue Playlist angelegt werden.  
Der Name darf nicht leer sein oder dem Platzhaltertext „Name der Playlist...“ entsprechen.  
Nach erfolgreicher Erstellung erscheint die neue Playlist sofort in der linken Liste.

**Übersicht und Funktionen**

**Linke Spalte:**

* Zeigt alle Playlists des Benutzers in einer Liste.
* Jede Playlist verfügt über:
  + **▶️-Button:** Startet die automatische Wiedergabe aller enthaltenen Medien.
  + **🗑️-Button:** Löscht die Playlist vollständig.

**Rechte Spalte:**

* Zeigt die enthaltenen Medien der aktuell ausgewählten Playlist.
* Für jedes Medium stehen folgende Funktionen zur Verfügung:
  + **▶️ Abspielen:** Startet das Medium direkt über die zentrale Wiedergabe.
  + **🗑️ Löschen:** Entfernt das Medium aus der Playlist (nicht aus der Mediathek).
  + **▲ / ▼:** Ändert die Reihenfolge der Medien in der Playlist.

Die Abspielreihenfolge der Medien wird in der Datenbank gespeichert und beim Löschen automatisch aktualisiert.

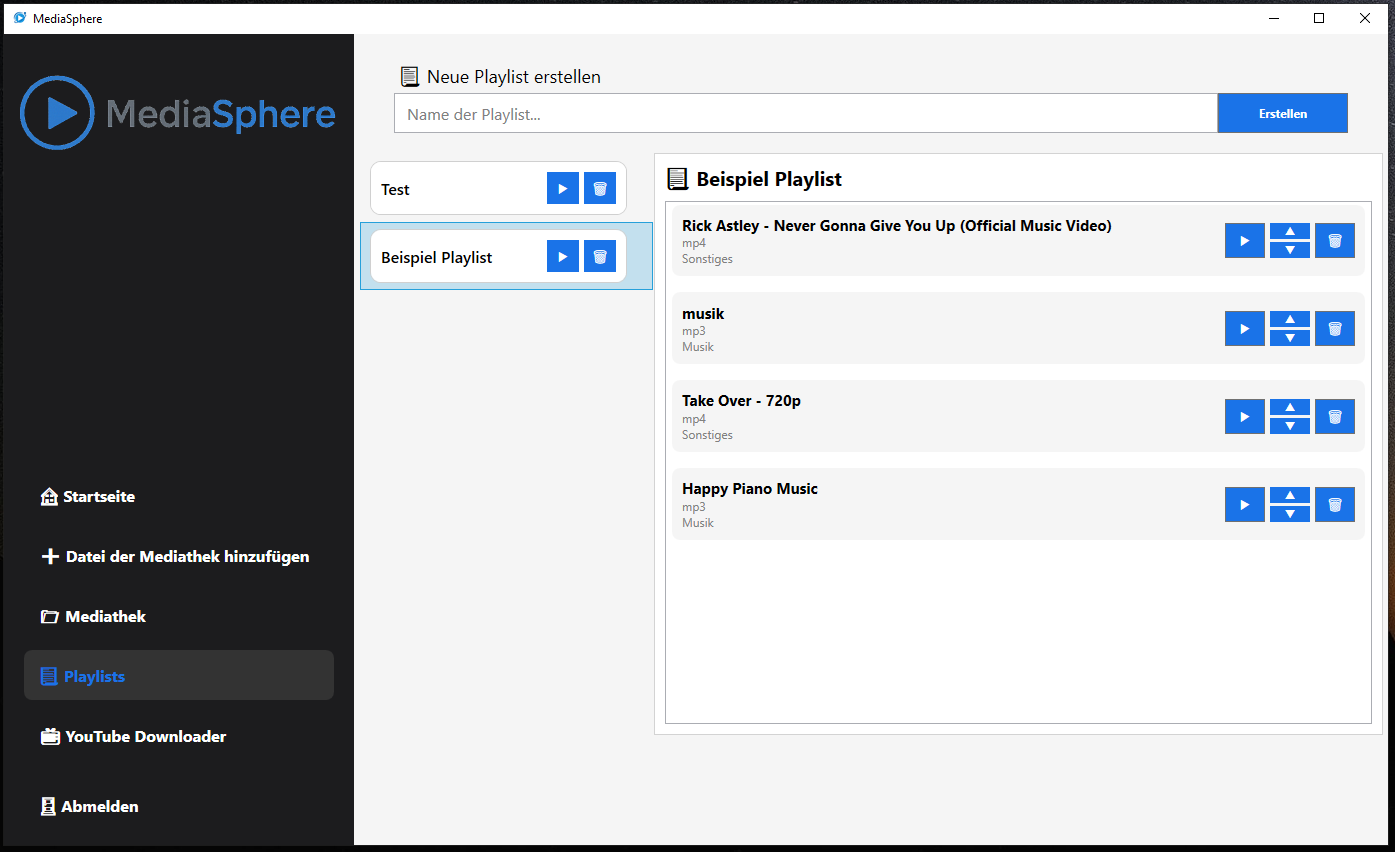


Abbildung 9: Playlist-Übersicht mit Funktionen zum Erstellen, Abspielen und Bearbeiten

## 2.7 Medienwiedergabe und Steuerung

Die Medienwiedergabe in MediaSphere ist zentral im Hauptfenster (MainWindow2.xaml) integriert. Sie unterscheidet zwischen Audio- und Videoformaten und stellt für beide Typen ein passendes Wiedergabefenster bereit:

* Für **Audiodateien** (z. B. .mp3) erscheint am unteren Rand der Anwendung eine Wiedergabeleiste mit Steuerungselementen.
* Für **Videodateien** (z. B. .mp4) wird ein zentriertes Overlay mit dunklem Hintergrund angezeigt, das die Wiedergabe in einem größeren Format ermöglicht.

**Hinweis:**  
Die Steuerungselemente (Play/Pause, Fortschrittsanzeige, Zeitangaben, Lautstärke und Loop-Modus) sind sowohl bei Audio- als auch bei Videowiedergabe verfügbar und funktionieren identisch. Sie sind lediglich visuell unterschiedlich platziert.

**Audiowiedergabe (untere Steuerleiste)**

Am unteren Rand der Anwendung erscheint eine Wiedergabeleiste. Sie enthält:

* **Aktuelle Zeit / verbleibende Zeit**: Links und rechts neben dem Fortschrittsbalken.
* **Fortschrittsanzeige (Slider)**: Ermöglicht das Vor- und Zurückspulen innerhalb der Audiodatei.
* **Titelanzeige**: Der aktuelle Titel wird links angezeigt.
* **Steuerungstasten**:
  + ⏮ Zurückspringen
  + ⏸/▶ Umschalten zwischen Wiedergabe und Pause (je nach aktuellem Zustand)
  + ⏭ Vorspringen
  + 🔁/🔃/🔂 Loop-Modus – zyklischer Wechsel zwischen: kein Loop, Playlist-Loop und Einzel-Loop.
* **Lautstärkeregler**: Rechts neben den Steuertasten.

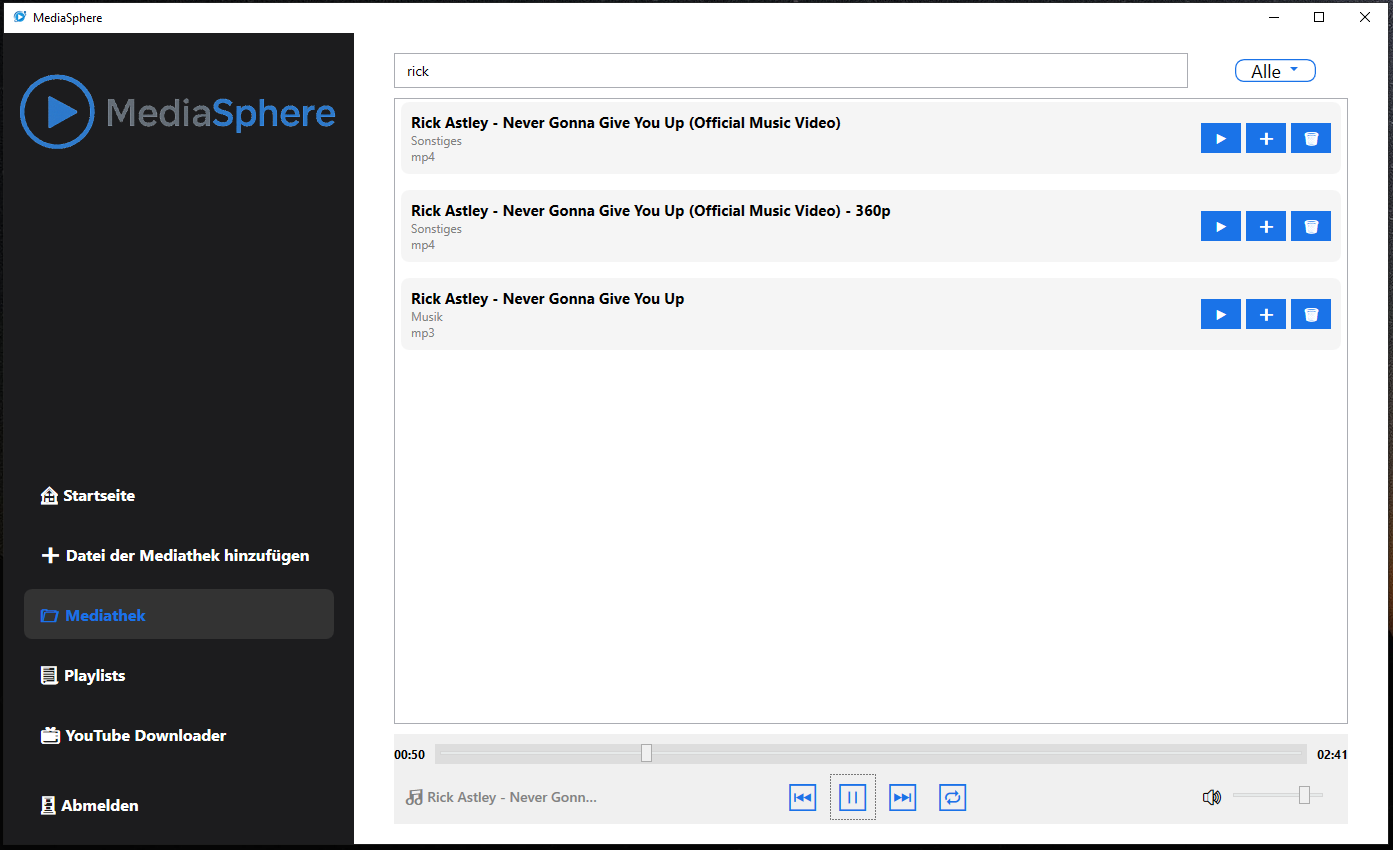


Abbildung 10: MediaSphere bei einer Audiowiedergabe

**Videowiedergabe (Overlay)**

Bei der Videowiedergabe wird ein Overlay über die gesamte Anwendung gelegt. Es enthält die gleichen Steuerelemente wie die Wiedergabeleiste für Audio, zusätzlich jedoch:

* **❌-Button**, um das Video zu schließen.
* Zentrierte Anzeige des Videos im Original-Seitenverhältnis (z. B. 16:9).



Abbildung 11: MediaSphere bei einer Videowiedergabe

## 2.8 YouTube Downloader

Der YouTube Downloader in MediaSphere ermöglicht das Herunterladen von YouTube-Videos oder -Audios direkt in die Mediathek. Die Funktion ist vollständig in die Anwendung integriert und bietet eine einfache Benutzeroberfläche zum Analysieren und Herunterladen von Inhalten über yt-dlp.

**Linkanalyse und Metadaten**

Der Nutzer kann einen beliebigen YouTube-Link in das Eingabefeld einfügen. Mit einem Klick auf **„🔍 Analysieren“** wird der Link überprüft und die Metadaten werden geladen.

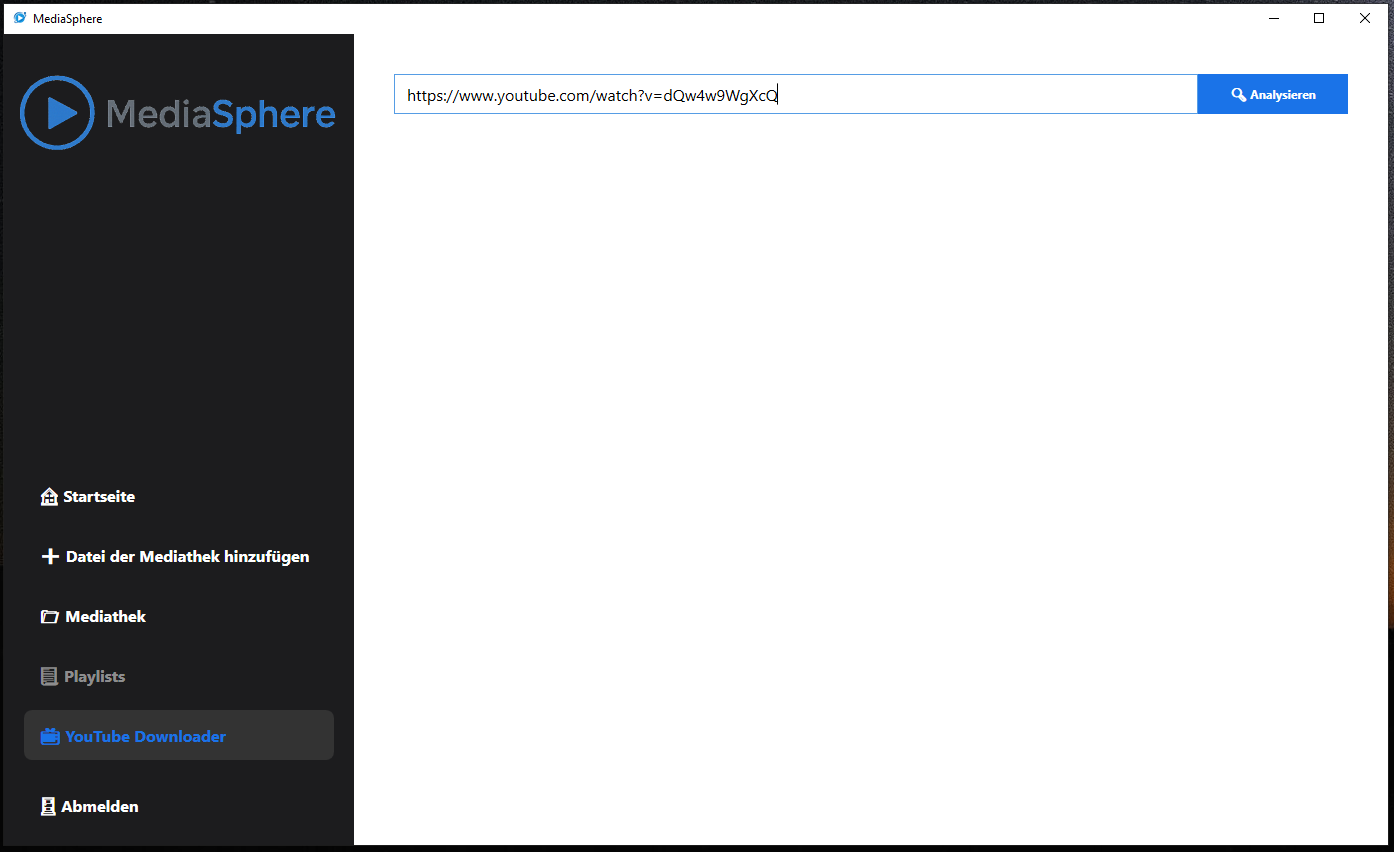


Abbildung 12: YouTube-Link wurde eingefügt, Analyse noch nicht gestartet.

Nach erfolgreicher Analyse erscheinen der Titel des Videos, die Möglichkeit zur Auswahl einer Kategorie sowie die gewünschte Auflösung.

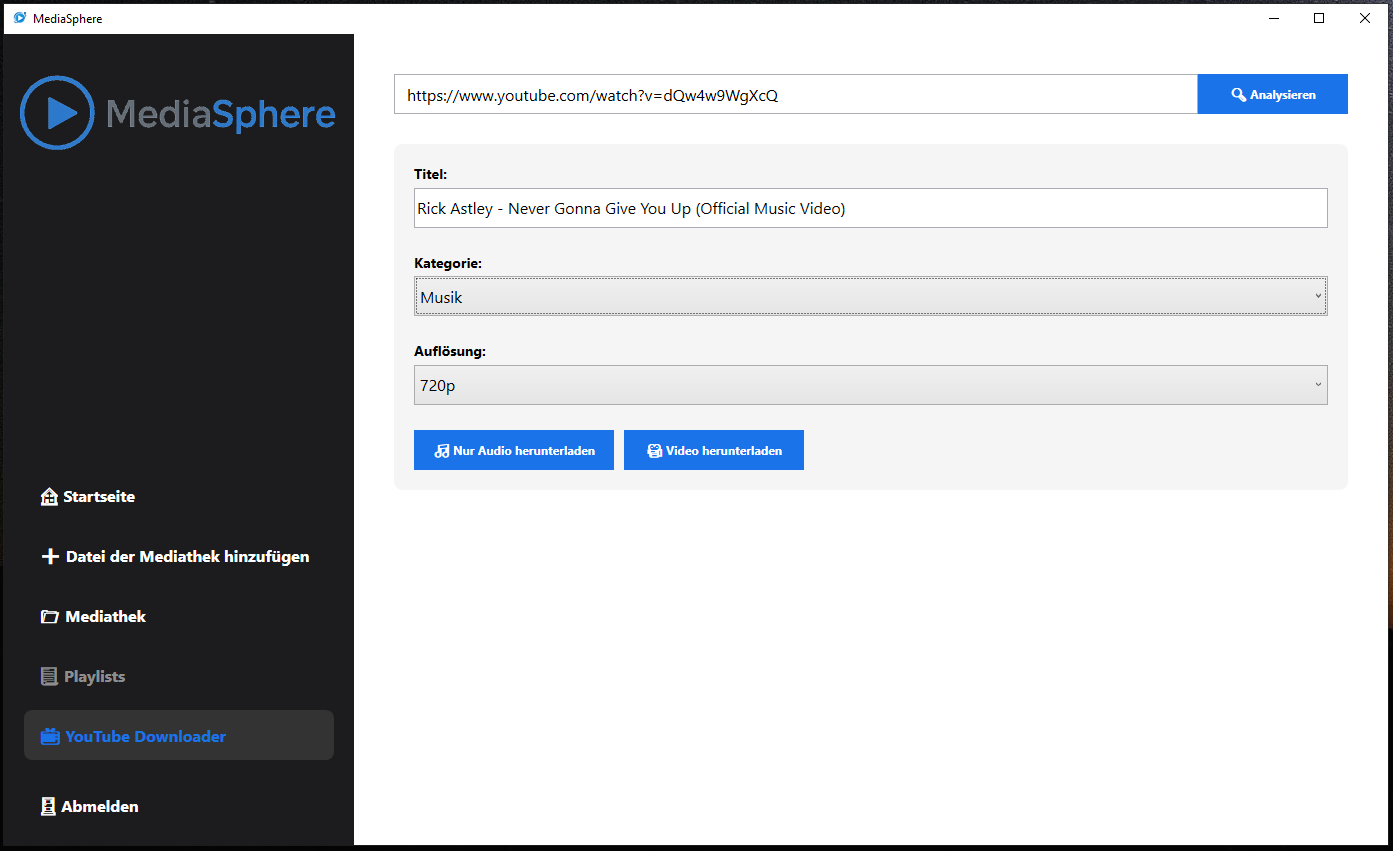


Abbildung 13: Erfolgreich analysiertes Video mit angezeigten Metadaten und auswählbaren Optionen.

**Download-Funktion**

Es stehen zwei Optionen zur Verfügung:

* **🎵 Nur Audio herunterladen:** Lädt die Audiospur des YouTube-Videos im MP3-Format herunter.
* **🎥 Video herunterladen:** Lädt das vollständige Video (im MP4-Format) in der ausgewählten Auflösung herunter.

Während des Downloads wird eine Lade-Animation angezeigt. Der Nutzer hat zusätzlich die Möglichkeit, den Download über den Button **„🛑 Download abbrechen“** zu stoppen.

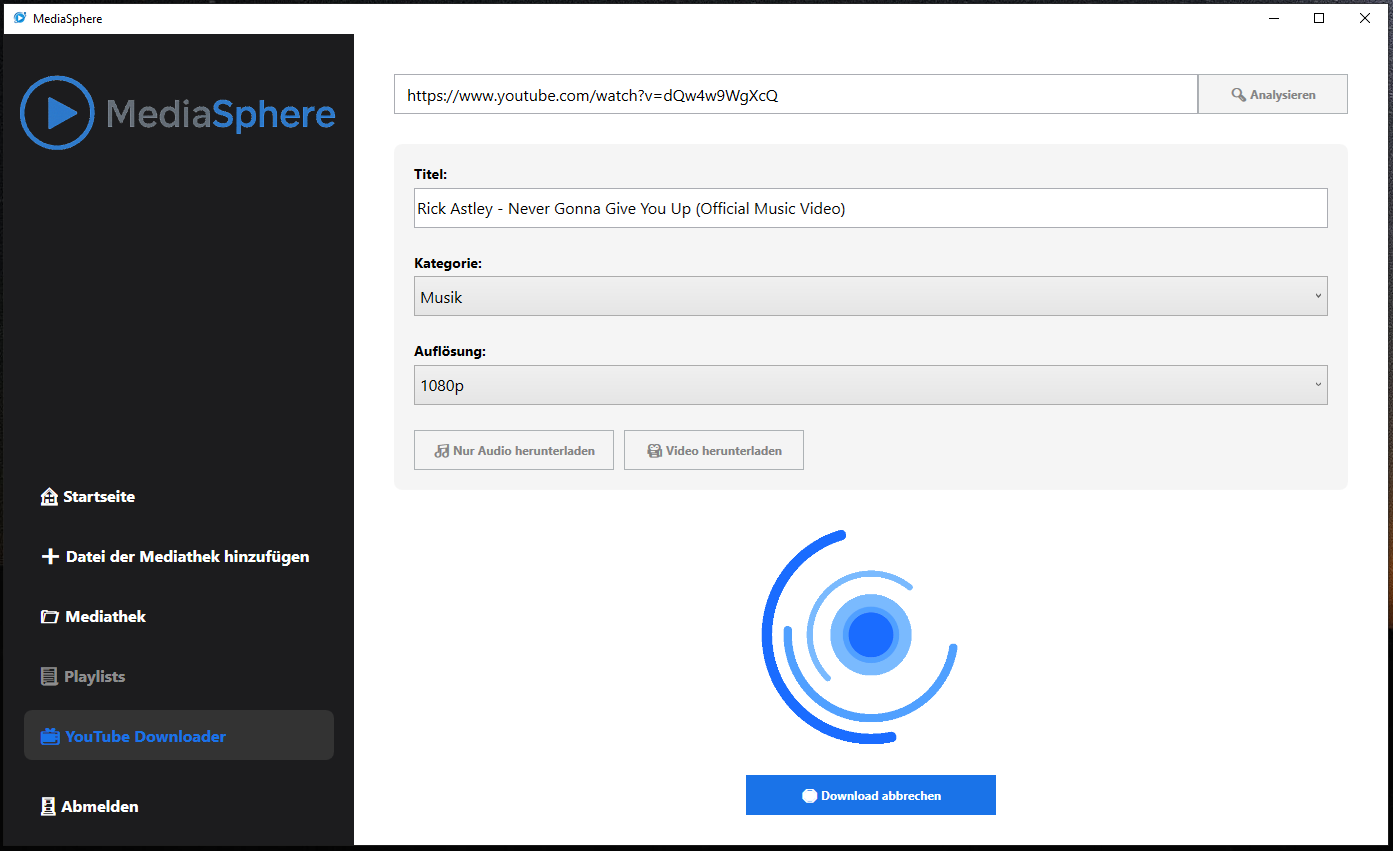


Abbildung 14: Laufender Download mit animierter Ladeanzeige und aktivem „🛑 Download abbrechen“-Button.

Nach dem Herunterladen wird die Datei automatisch in den Mediathek-Ordner kopiert und mitsamt Pfad, Typ, Titel und gewählter Kategorie in die Datenbank eingetragen. Existiert bereits eine gleichnamige Datei, wird sie durch eine fortlaufende Nummerierung (z. B. „(1)“, „(2)“ usw.) eindeutig benannt. Zum Abschluss erscheint eine kurze Bestätigung über den erfolgreichen Download.

# 3. Implementierung

## 3.1 Überblick über die Projektstruktur

Die Anwendung *MediaSphere* wurde mit dem Framework **Windows Presentation Foundation (WPF)** unter Verwendung von **C# und .NET 8** entwickelt. WPF ermöglicht eine moderne, flexible Benutzeroberfläche mit Unterstützung für XAML, wodurch komplexe Layouts und visuelle Designs effizient umgesetzt werden können. Als Entwicklungsumgebung wurde **Visual Studio 2022** verwendet.

Die Anwendung ist als klassische Desktopanwendung konzipiert und arbeitet vollständig lokal. Zur Datenspeicherung kommt **SQLite** als leichtgewichtige relationale Datenbank zum Einsatz. Hier werden unter anderem Benutzerinformationen, Mediendateien und Playlists verwaltet. Für die Erweiterung der Funktionalität um einen YouTube-Downloader wurde das Tool **yt-dlp** integriert, das per Kommandozeilenaufruf aus der Anwendung heraus genutzt wird.

Der Aufbau der Anwendung folgt einer **modularen Struktur**: Die Benutzeroberfläche ist in einzelne Seiten (**UserControls**) unterteilt. Diese werden abhängig vom Programmzustand in eines von zwei Hauptfenstern eingebettet. Beim Start der Anwendung wird zunächst das Fenster **MainWindow.xaml** geöffnet, in dem die Login- und Registrieransichten als UserControls geladen werden. Nach erfolgreicher Anmeldung oder Auswahl des Gast-Modus wird das zentrale Fenster **MainWindow2.xaml** geöffnet, in dem alle Hauptfunktionen zur Medienverwaltung und -wiedergabe zur Verfügung stehen.

Innerhalb von MainWindow2.xaml erfolgt die Navigation über eine linke Seitenleiste, über die die verschiedenen Bereiche wie Startseite, Mediathek, Playlists, Datei hinzufügen und YouTube Downloader erreicht werden können. Die Medienwiedergabe erfolgt entweder über eine **untere Steuerleiste** für Audiodateien oder über ein **zentral platziertes Video-Overlay**, das bei Videodateien eingeblendet wird. Die Steuerungselemente (Play, Pause, Lautstärke, Fortschritt, Loop, Skip-Funktionen) sind für beide Medientypen vollständig implementiert.

Zusätzlich bietet die Anwendung ein **Login-System** mit Registrierung, Gast-Modus und der Möglichkeit, den Anmeldestatus lokal zu speichern („Angemeldet bleiben“). Jeder registrierte Benutzer kann eigene Playlists anlegen, verwalten und speichern, während Gäste nur grundlegende Wiedergabefunktionen nutzen können.

## 3.2 Klassenstruktur und Datenmodell

Das Datenmodell der Anwendung *MediaSphere* ist eng mit der zugrunde liegenden SQLite-Datenbank verknüpft und bildet deren Tabellen in Form von einfachen C#-Klassen ab. Jede dieser Klassen steht für eine Entität innerhalb der Anwendung und wird zur Laufzeit zur Verwaltung und Anzeige von Daten verwendet.

### 3.2.1 Medium

Die Klasse Medium repräsentiert eine Mediendatei – entweder eine Audiodatei im MP3-Format oder eine Videodatei im MP4-Format. Jede Datei besitzt einen eindeutigen Pfad sowie Metadaten wie Titel, Typ und Kategorie. Diese Informationen entsprechen den Einträgen in der Tabelle Medien der Datenbank.

public class Medium

{

public int MedienID { get; set; }

public string Pfad { get; set; }

public string Typ { get; set; }

public string Titel { get; set; }

public string Kategorie { get; set; }

}

### 3.2.2 Playlist

Die Klasse Playlist dient zur Verwaltung benutzerdefinierter Wiedergabelisten. Jede Playlist gehört genau einem Benutzer und kann beliebig viele Medien enthalten. Zusätzlich speichert sie den Namen sowie das Erstellungsdatum. Diese Klasse entspricht der Tabelle Playlist in der Datenbank.

public class Playlist

{

public int PlaylistID { get; set; }

public int BenutzerID { get; set; }

public string Name { get; set; }

public DateTime Erstellungsdatum { get; set; }

public override string ToString()

{

return Name;

}

}

### 3.2.3 PlaylistMedium

Die Klasse PlaylistMedium stellt eine Zuordnungstabelle zwischen Playlists und Medien dar. Sie bildet eine **n:m-Beziehung** ab, indem sie angibt, welche Mediendatei in welcher Playlist enthalten ist. Zusätzlich wird mit dem Attribut Reihenfolge die Wiedergabereihenfolge innerhalb der Playlist gespeichert. Diese Struktur wird direkt auf die Tabelle PlaylistMedien abgebildet.

public class PlaylistMedium

{

public int PlaylistID { get; set; }

public int MedienID { get; set; }

public int Reihenfolge { get; set; }

}

### 3.2.4 Klassendiagramm

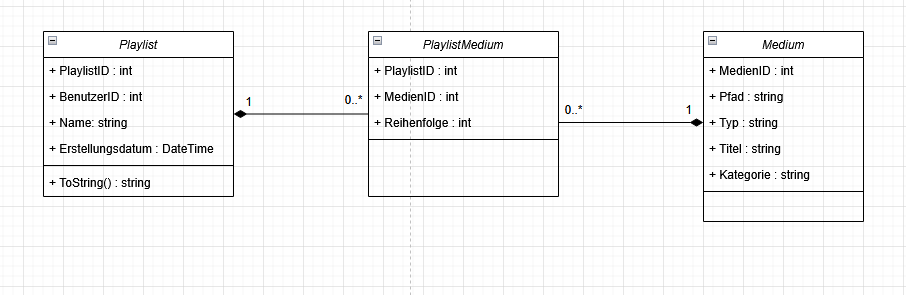


Abbildung 15: Klassendiagramm von MediaSphere

## 3.3 Datenbankanbindung

Die Anwendung *MediaSphere* verwendet **SQLite** als lokale Datenbanklösung, um Benutzerdaten, Mediendateien, Playlists und deren Zuordnungen effizient und dauerhaft zu speichern. Die Anbindung erfolgt direkt über die Klasse **SQLiteConnection** aus dem .NET-Systemnamespace **System.Data.SQLite**. Dadurch kann vollständig auf ein externes Datenbankmanagementsystem verzichtet werden, was die Portabilität der Anwendung erhöht.

Beim Start der Anwendung wird geprüft, ob der Anwendungsordner sowie die Datenbankdatei MediaSphere.db bereits im AppData-Verzeichnis des aktuellen Benutzers existieren. Falls nicht, werden sie automatisch erstellt. Dies erfolgt im Konstruktor von MainWindow.xaml, dem Startfenster der Anwendung.

Die folgende Codepassage definiert die relevanten Pfade und die SQLite-Verbindungszeichenfolge:



Abbildung 16: Definition des Anwendungsordners und der SQLite-Datenbankdatei im AppData-Verzeichnis von Windows

Wird beim Programmstart keine Datenbankdatei gefunden, wird diese durch SQLiteConnection.CreateFile() erzeugt. Anschließend ruft die Anwendung die Methode CreateTables() auf, welche die für den Betrieb erforderlichen Tabellen mit SQL-Befehlen erstellt.

Zunächst wird der Befehl PRAGMA foreign\_keys = ON ausgeführt, um die Einhaltung von Fremdschlüsselbeziehungen zu gewährleisten. Danach werden vier Tabellen erstellt:

* **Benutzer:** enthält Benutzername und verschlüsseltes Passwort
* **Medien:** speichert Pfad, Typ, Titel und Kategorie jeder Mediendatei
* **Playlist:** enthält benutzergebundene Wiedergabelisten
* **PlaylistMedien:** verbindet Medien und Playlists und speichert deren Reihenfolge

Ein Beispiel für eine der SQL-Anweisungen:

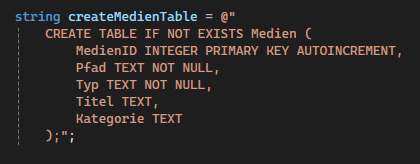
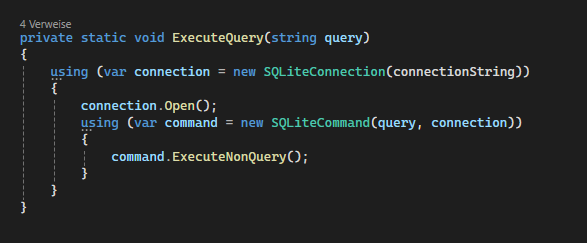


Abbildung 17: SQLite-Befehl zum Anlegen der Tabelle "Medien"

Die Methode ExecuteQuery() übernimmt dabei die Ausführung der SQL-Befehle:



Diese Initialisierungslogik stellt sicher, dass die SQLite-Datenbank beim ersten Start korrekt aufgebaut wird und alle erforderlichen Tabellen zur Verfügung stehen. Gleichzeitig wird bei jedem Programmstart sichergestellt, dass die Datenbankstruktur vorhanden und konsistent ist.

## 3.4 Login und Registrierung

Beim Start von *MediaSphere* erscheint zunächst das Login-Fenster, in dem sich ein Benutzer anmelden, registrieren oder als Gast fortfahren kann. Die Benutzerverwaltung ist vollständig in die Anwendung integriert und basiert auf einer lokalen SQLite-Datenbank. Die Umsetzung erfolgt über zwei modular aufgebaute UserControls: **Login.xaml** und **Registrieren.xaml**, die dynamisch im Hauptfenster **MainWindow.xaml** geladen werden. Die optische Gestaltung und Benutzerführung sind bereits im Benutzerhandbuch (siehe **Abbildung 3 und 4**) dargestellt. Im Folgenden wird die technische Umsetzung beschrieben.

### 3.4.1 Login-Funktion

Die Login-Ansicht besteht aus zwei Eingabefeldern (Benutzername, Passwort), einer Checkbox („Angemeldet bleiben“) sowie Buttons zur Anmeldung, zum Gastzugang und zur Weiterleitung zur Registrierung. Die Logik befindet sich in der Datei Login.xaml.cs.

Beim Klick auf den Anmelden-Button wird die Methode AnmeldedatenÜberprüfen() aufgerufen. Sie vergleicht den eingegebenen Benutzernamen und das verschlüsselte Passwort mit den Daten in der Benutzer-Tabelle der SQLite-Datenbank. Die Passwortprüfung erfolgt über eine SHA256-Hashfunktion, die aus Sicherheitsgründen keine Klartextpasswörter speichert.

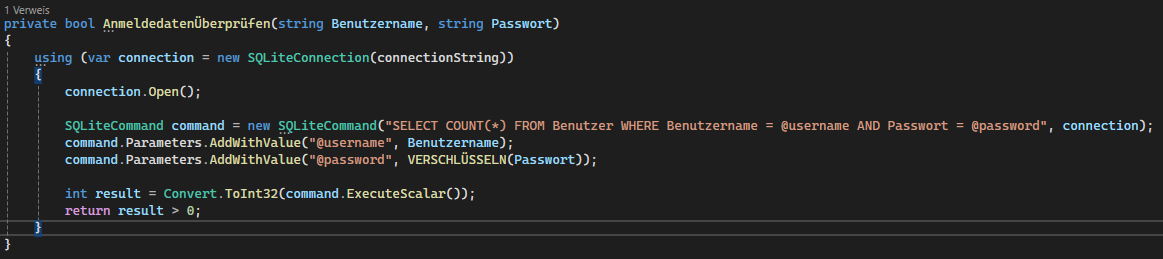


Abbildung 18: Anmeldedatenprüfung mit Passwort-Hashing und SQL-Abfrage

Wird die Anmeldung erfolgreich durchgeführt, wird geprüft, ob die Option „Angemeldet bleiben“ aktiv ist. In diesem Fall wird der Benutzername dauerhaft in den Anwendungseinstellungen gespeichert und beim nächsten Start automatisch verwendet. Danach wird das Hauptfenster MainWindow2.xaml mit dem eingeloggten Benutzer gestartet.

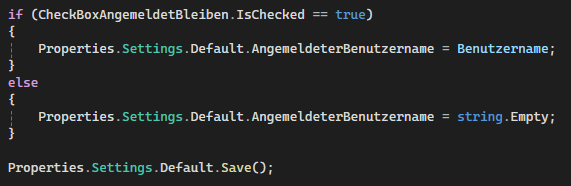


Abbildung 19: Umsetzung der „Angemeldet bleiben“-Funktion durch Speichern des Benutzernamens in Properties.Settings

Alternativ kann sich der Benutzer über den Button **„Als Gast fortfahren“** anmelden. In diesem Fall wird „Gast“ als Benutzername gesetzt und der Zugriff auf Funktionen wie Playlists deaktiviert. Auch hier wird direkt das Hauptfenster geladen.

### 3.4.2 Registrierung

Wird der „Registrieren“-Button geklickt, wird das UserControl Registrieren.xaml geladen. Dieses enthält drei Eingabefelder: Benutzername, Passwort, Passwort-Wiederholung. Ein Button zur Rückkehr zum Login ist ebenfalls vorhanden. Die Validierung prüft in Echtzeit, ob die Passwörter übereinstimmen und ob alle Felder ausgefüllt sind. Nur dann wird der **Registrieren-Button aktiviert**.

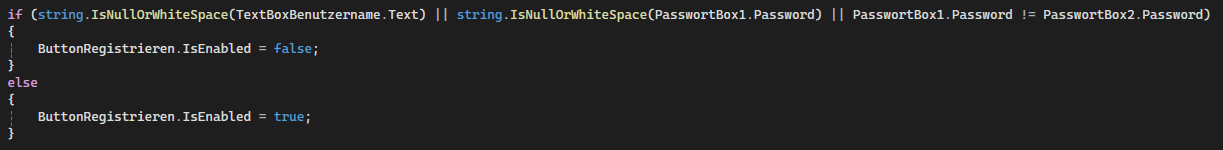


Abbildung 20: Validierung der Eingabefelder zur Aktivierung des „Registrieren“-Buttons nur bei vollständiger und korrekter Eingabe

Nach Klick auf „Registrieren“ wird geprüft, ob der Benutzername bereits existiert. Falls nicht, wird der Benutzer zusammen mit dem verschlüsselten Passwort in die Datenbank eingetragen. Anschließend erscheint eine Erfolgs- oder Fehlermeldung in einem benutzerdefinierten Dialogfenster (CustomDialog).

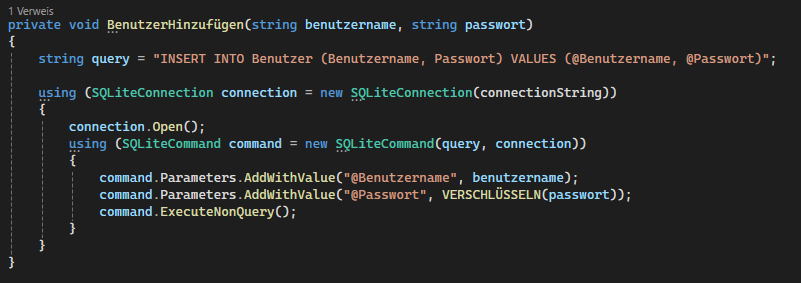


Abbildung 21: Einfügen eines neuen Benutzers in die Benutzer-Tabelle der SQLite-Datenbank mit verschlüsseltem Passwort

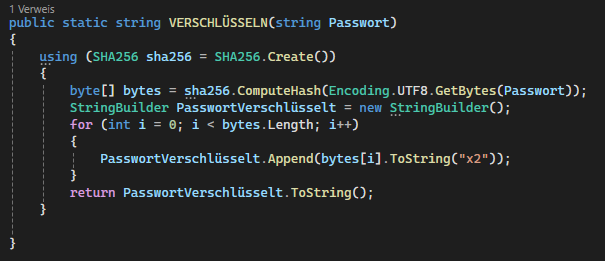


Abbildung 22: Passwortverschlüsselung mittels SHA256-Hashfunktion vor Speicherung in der Datenbank

Die gesamte Benutzerverwaltung ist lokal umgesetzt und benötigt keine Internetverbindung oder externe Authentifizierung. Durch die Möglichkeit, sich als Gast anzumelden, kann die Anwendung auch ohne Konto genutzt werden, wenn nur grundlegende Funktionen wie das Abspielen von Dateien benötigt werden.