

# MP3-Player

von Jasmin Sölling, Julian Bermel, Darius Keil,  
Timo Seuser und Jennifer Ried

# **Inhaltsangabe**

1. Abbildungsverzeichnis
2. Einleitung
  - 2.1. Design
3. Funktionalitäten
  - 3.1. Playliste
  - 3.2. Anzeige Playliste
  - 3.3. Lieder hinzufügen
  - 3.4. Suche
  - 3.5. Wiedergabe
  - 3.6. Weiter/Zurück
  - 3.7. Vorspulen/Zurückspulen
  - 3.8. Lautstärke
  - 3.9. Zufall/Reihenfolge ändern
  - 3.10. Löschen einzelner Lieder
  - 3.11. Speichern der Playlist
4. Diagramme
  - 4.1. 4+1 Sichtenmodell
    - 4.1.1. Klassendiagramm
    - 4.1.2. Aktivitätsdiagramm
    - 4.1.3. Komponentendiagramm
    - 4.1.4. Verteilungsdiagramm
    - 4.1.5. Anwendungsfalldiagramm
  - 4.2. Datenflussdiagramm
5. Quellen

# 1. Abbildungsverzeichnis

Abbildung 1: Entwurf des MP3-Players .....	3
Abbildung 2: End Design des Mp3-Players.....	4
Abbildung 3: Lieder hinzufügen.....	6
Abbildung 4: Aufruf der Suche.....	6
Abbildung 5: Suchwidget.....	6
Abbildung 6: Schaltfläche.....	7
Abbildung 7: Lautstärke.....	8
Abbildung 8: Stumm.....	8
Abbildung 9: Aus der Playliste entfernen.....	9
Abbildung 10: Klassendiagramm.....	10
Abbildung 11: Aktivitätsdiagramm.....	11
Abbildung 12: Komponentendiagramm.....	12
Abbildung 13: Verteilungsdiagramm.....	13
Abbildung 14: Anwendungsfalldiagramm.....	14
Abbildung 15: Datenflussdiagramm.....	15

## 2. Einleitung

Bei der zweiten Aufgabe des Moduls "User Interface Design" ging es darum einen MP3-Player zu implementieren, der eine Anzeige der Playlist sowie verschiedene Funktionalitäten wie das Hinzufügen und Entfernen von Liedern, Starten, Stoppen, Pausieren, Vor- und Zurückspulen des aktuellen Liedes und das Vor- und Zurückspringen zum vorherigen oder nächsten Lied. Die genaue Aufgabe entnehmen sie der Datei "aufgabe\_2\_29\_10\_2021\_12 06 13".

### 2.1 Design

Zu Beginn des Projektes wurde ein Entwurf zur groben Übersicht und Anordnung der Elemente erstellt, der folgendermaßen aussah:



Abbildung 1: Entwurf des MP3-Players

Auf der Basis dieses Entwurfs ist die weitere Implementierung erfolgt. Jedoch gab es noch leichte Änderungen während der Entwicklung, sodass das Programm am Ende folgendermaßen aussieht:

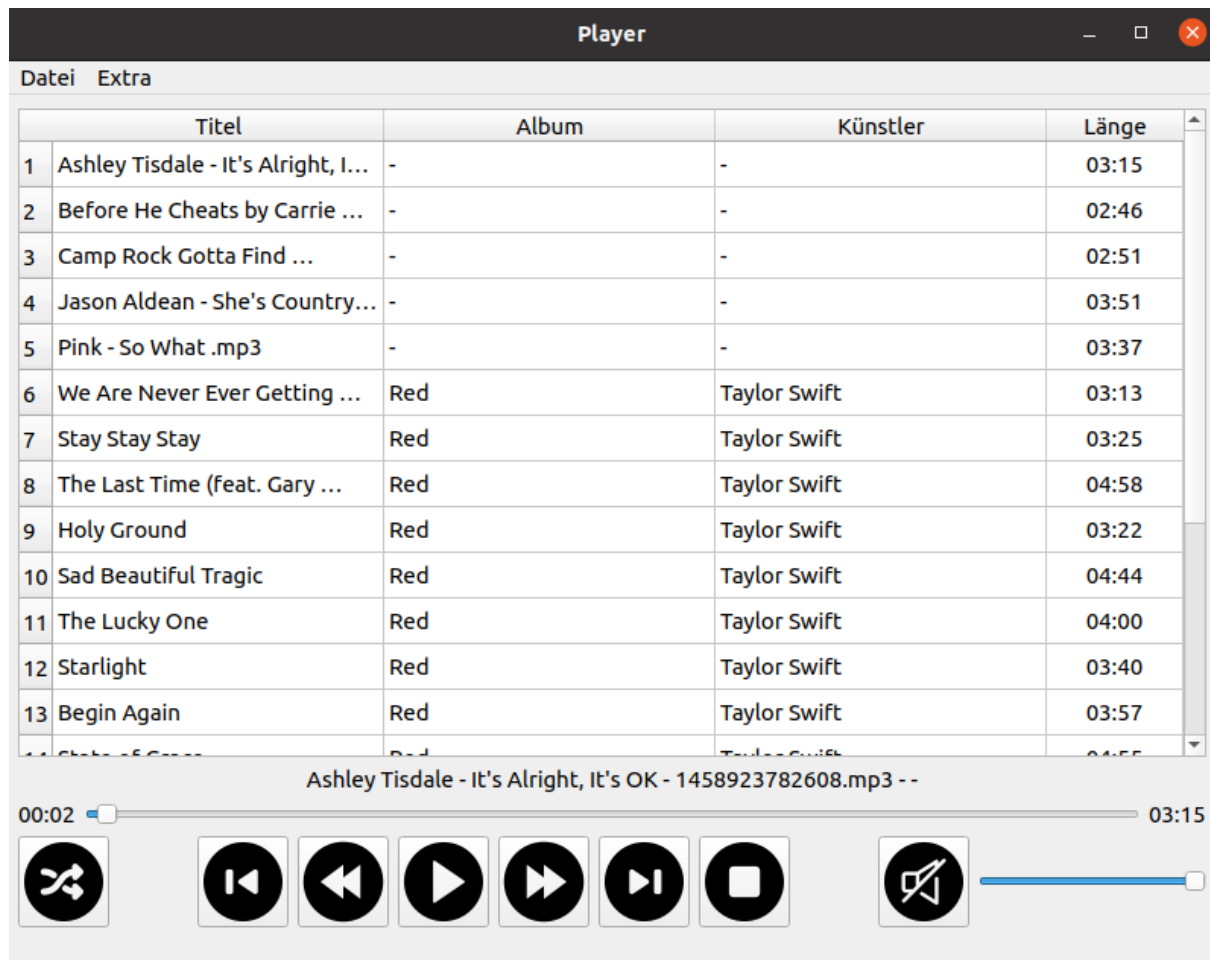


Abbildung 2: End Design des Mp3-Players

Hier ist zusehen, dass die grobe Platzierung der Elemente übernommen wurde, es jedoch noch leichte Änderungen gibt. Zum Beispiel wurde die Reihenfolge der Button so geändert, dass am Wiedergabe-Schalter die Button am nächsten sind, die das Lied vor- und zurückspringen und erst dann die Buttons, die in der Playlist vor- und zurückspringen. Zudem wurde noch ein Label mit Titel und Künstler des aktuellen Liedes sowie Labels zur Darstellung der verstrichenen Zeit und der Gesamtzeit des aktuellen Liedes eingefügt. Das Farbkonzept wurde in tristen schwarz und weiß Tönen gehalten, da hier keine Farbe nötig ist. Die Sliders haben einen Blauton bekommen, um die Aufmerksamkeit des Benutzers auf sich zu ziehen, dass sie hier etwas ändern können. Zudem wird die Hintergrundfarbe des Stumm- und Zufallsbutton auch blau, wenn diese getätigt werden.

## 3. Funktionalitäten

Im Folgenden werden die einzelnen Funktionalitäten des Mp3-Players beschrieben, sowohl wie der Anwender diese zu bedienen hat und welche Ergebnisse diese mit sich bringen, als auch ihre technische Umsetzung. Jedes Schaltelement hat eine eigene Kurzinfo beim darüber Hovern mit der Maus, um auch im Zweifel anzeigen zu können, welche Funktion dieses Schaltelement verfolgt.

Bei der Entwicklung wurde die offizielle Qt Dokumentation genutzt, um herauszufinden welche Signals, Slots und Funktionen die einzelnen Klassen besitzen. Zudem wurde bei Problemen häufig Lösungsansätze von der Website Stackoverflow oder dem Qt-Forum verwendet.

### 3.1 Playliste

Die Playlist wurde über die Klasse QMediaPlayer realisiert. Um diese Klasse für dieses Projekt nutzen zu können wurde das Paket "QtMultimedia5" verwendet, das über folgenden Befehl auf die zur Verfügung gestellten virtuellen Maschinen mit Ubuntu-Image installiert werden kann : "sudo apt-get install qtmultimedia5-dev". Gegebenenfalls muss ebenfalls ein Plugin Paket installiert werden mit dem Befehl: sudo apt-get install libqt5multimedia5-plugins. Der Player ist mit einer Playlist der Klasse QMediaPlaylist verbunden.

### 3.2 Anzeige Playliste

Für die Anzeige der Playlist wurde ein QTableWidget verwendet. Eine Zeile repräsentiert dabei ein Lied. Von jedem Lied werden der Titel, das Album und der/die Künstler/-in angezeigt unter der Voraussetzung, dass diese Daten bei der mp3-Datei hinterlegt sind. Ist kein Titel hinterlegt so wird der Name der Datei als Titel genutzt. Es gibt zwei Möglichkeiten die Reihenfolge der Playlist zu verändern. Zum Einem durch das Ziehen einzelner Lieder über die Zeilennummer durch Drag and Drop. Zum Anderen durch das alphabetische Sortieren der Spalten durch das Anklicken der Spaltenüberschriften.

### 3.3 Lieder hinzufügen

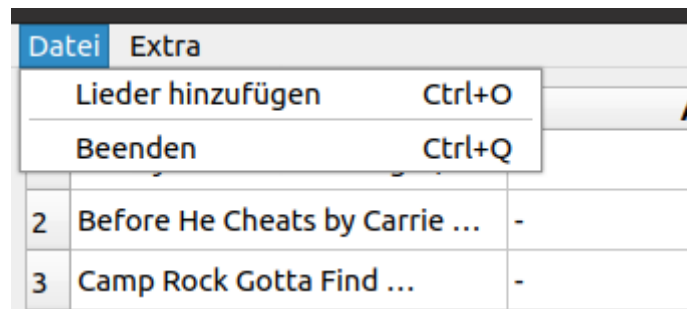


Abbildung 3: Lieder hinzufügen

Ein Lied kann über verschiedene Wege hinzugefügt werden. Es kann der Shortcut "Strg + o" oder die Menübar unter Datei genutzt werden, um den Datei Dialog zu öffnen. Hier können ein oder mehrere Lieder ausgewählt werden. Diese werden dann der Playlist hinzugefügt. Dabei wird jedes Lied noch einem temporären Player hinzugefügt, um über diesen Player auf die Daten, also Titel, Album und Künstler, der mp3 zugreifen zu können. Sind keine Daten bei der mp3-Datei hinterlegt wird der Dateiname als Titel benutzt und alle weiteren Felder werden mit "-" befüllt.

### 3.4 Suche

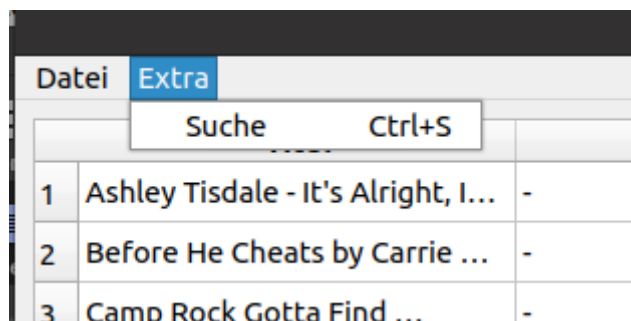


Abbildung 4: Aufruf der Suche

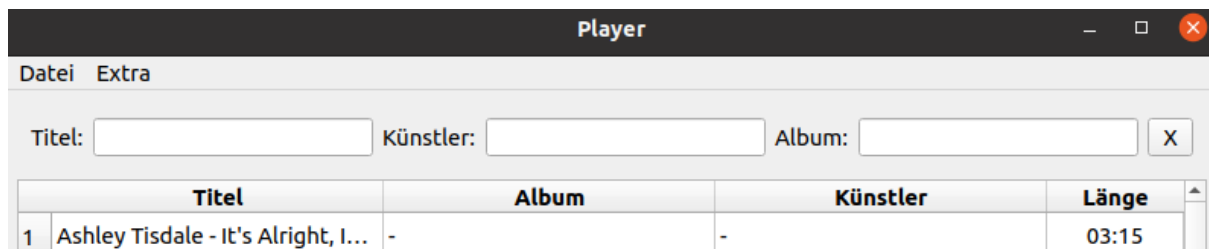




Abbildung 5: Suchwidget

Es kann nach den drei Spalten-Kategorien Titel, Album und Künstler gesucht werden. Bei der Suche handelt es sich um eine interaktive Suche, das heißt sobald ein Buchstabe in eines der drei Suchfelder eingegeben wird, wird nach diesem in den jeweiligen Feldern der Spalte gesucht. Wird ein Suchfeld leer gelassen, wird es

interpretiert als würde der Benutzer nach nichts suchen und so wird keine Suche durchgeführt. Es erscheinen analog zur Eingabe, die zur Eingabe passenden Lieder in der Liste. Bei der Eingabe wird auf Groß- und Kleinschreibung geachtet.

### 3.5 Wiedergabe

Sobald mindestens ein Lied der Playlist hinzugefügt wird, wird die Wiedergabe gestartet. Dabei ändert sich das Symbol des Wiedergabe-Pause-Button in ein

Pause-Symbol  um zu signalisieren, dass bei erneuten Klicken auf diesen Button das Lied pausiert wird. Genauso wird bei Pausierung das Symbol des Buttons in das Wiedergabe-Symbol  geändert. Es kann auch ein Lied durch Doppelklick auf die jeweilige Zeile in der Tabelle ausgewählt und damit wiedergegeben werden. Auch kann das Pausieren oder Wiedergeben durch einen Shortcut über die Leertaste durchgeführt werden. Zudem gibt es eine Anzeige, die den Fortschritt des Liedes widerspiegelt. Zusätzlich befindet sich links neben dieser Anzeige die genaue Angabe der schon abgespielten Zeit und rechts daneben die Gesamtlänge des aktuell laufenden Liedes.

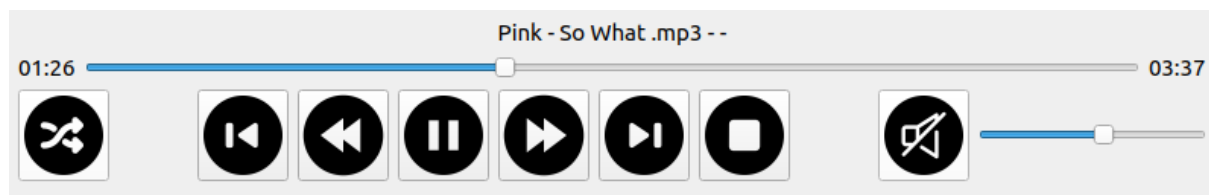






Abbildung 6: Schaltfläche

### 3.6 Weiter / Zurück

Wenn der Weiter-Button  oder der dazugehörige Shortcut mit der Pfeil nach rechts Taste betätigt werden, wird das nächste Lied der Playlist wiedergegeben. Bei dem Zurück-Button  oder dem dazugehörigen Shortcut über die Pfeil nach links Taste wird das vorherige Lied in der Playlist wiedergegeben. Die Funktionalität konnte durch die öffentlichen Slots “next” und “previous” realisiert werden. Zudem wird nach betätigen des Buttons das jeweilige Lied wiedergeben.

### 3.7 Vorspulen / Zurückspulen

Durch Anklicken des Vorspulen-Buttons , wird das aktuell wiedergegebene Lied vorgespult. Analog kann das Zurückspulen durch das Betätigen des

Zurückspulen-Buttons  realisiert werden. Durch das Klicken des Buttons wird das Lied um 5 Sekunden nach vorne- oder zurückgespult. Dies wird durch das



Hinzufügen oder Abziehen von 5 Sekunden von der aktuellen Zeit realisiert. Dabei kann beim Vorspulen auch zum nächsten Titel gesprungen werden, jedoch kann beim Zurückspulen nicht das Lied gewechselt werden.

### 3.8 Lautstärke

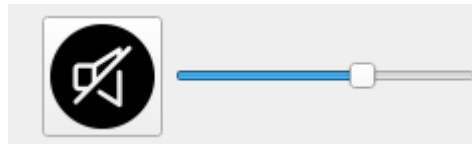


Abbildung 7: Lautstärke

Die Lautstärke wurde durch einen Slider realisiert, der sich unten rechts im Fenster befindet. Dieser kann durch anklicken und verschieben beeinflussen werden. Wenn der Stumm-Button oder der Shortcut "Strg + m" getätigt wird, wird die Lautstärke auf null gesetzt.

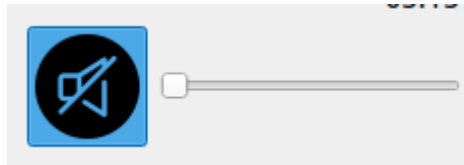




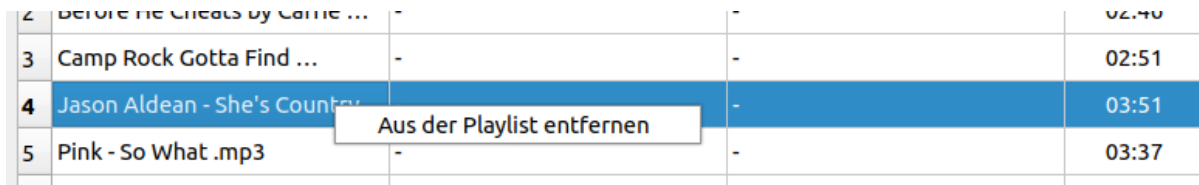
Abbildung 8: Stumm

Durch erneutes betätigen des Buttons oder des Shortcuts wird die Lautstärke auf den vorherigen Wert gesetzt. Durch die Pfeiltaste nach unten wird die Lautstärke heruntergesetzt. Analog dazu erhöht sich die Lautstärke mit der Pfeiltaste nach oben. Durch dauerhaftes Drücken der Taste kann die Lautstärke schneller erhöht oder reduziert werden.

### 3.9 Zufall / Reihenfolge ändern

Die Playlist kann zufällig abgespielt werden, indem der Zufallsbutton  oder "Strg + r" als Shortcut verwendet wird. Dadurch wird in der Playlist der Modus auf Zufall gestellt und der Zufallsbutton blau markiert . Die zufällige Wiedergabe kann durch erneutes Betätigen des Zufallsbutton wieder beendet werden.

### 3.10 Löschen einzelner Lieder



The image shows a table with four rows. The fourth row is highlighted in blue. A context menu is open over the fourth row, displaying the text 'Aus der Playlist entfernen'. The table has four columns: an index column, a title column, a duration column, and an empty column.

2	Before He Cheats by Carrie ...	-	02:40
3	Camp Rock Gotta Find ...	-	02:51
4	Jason Aldean - She's Country	-	03:51
5	Pink - So What .mp3	-	03:37

Abbildung 9: Aus der Playliste entfernen

Durch Rechtsklick auf eine Zeile der Tabelle kann ein Element aus der Playlist entfernt werden. Hierbei wird ein ContextMenu unterhalb des Mauszeigers erscheinen mit der Option: "Aus der Playlist entfernen". Durch Anklicken dieses Feldes wird die ausgewählte Zeile sowohl aus der Playlist als auch aus der Tabelle entfernt.

### 3.11 Speichern der Playlist

Die in der Playlist hinzugefügten Lieder werden in einer json-Datei durch ihre Url gespeichert. Bei jedem Hinzufügen eines Liedes oder Löschen eines Liedes wird die json-Datei neu gespeichert, indem die Url jedes einzelnen Liedes der Playlist abgerufen wird. Dies sorgt auch dafür, dass die Reihenfolge beim letzten Löschen oder Hinzufügen selbst nach Schließen des Programmes noch erhalten bleibt. Beim Start des Programms wird die zuvor gespeicherte Json Datei aufgerufen und die Playliste ist wie zuvor beim Schließen wieder vorhanden und wird direkt wiedergegeben.

## 4. Diagramme

### 4.1 4+1 Sichtenmodell

Das 4+1 Sichtenmodell nach Kruchten ist ein Model, um die verschiedenen Sichten auf ein Softwaresystem darzustellen.

#### 4.1.1 Klassendiagramm

Ein Klassendiagramm veranschaulicht alle Klassen, welche das Programm benutzt und zeigt die logische Sicht des 4+1 Sichtenmodells.

Es gibt eine Klasse, diese enthält die Informationen der MP3-Dateien. Darunter fallen Titel, Künstler, Album, Länge des Songs und die URL, die den Speicherort der Datei angibt.

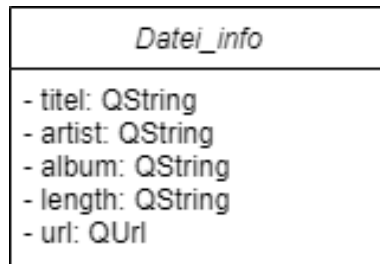


Abbildung 10: Klassendiagramm

## 4.1.2 Aktivitätsdiagramm

Ein Aktivitätsdiagramm veranschaulicht alle Prozessabläufe, welche das Programm ausführen kann und zeigt die Prozesssicht des 4+1 Sichtenmodells.

Bei dem Start des Programms wird als erstes die Benutzeroberfläche geöffnet. Von hier aus können acht verschiedene Prozessabläufe ausgeführt werden. Dazu gehören zum Beispiel "MP3 hinzufügen", "Wiedergabe Stopp" oder auch "Stumm schalten". Nachdem einer dieser Abläufe angefangen wurde, wird nach einer weiteren Aktion der Prozess erfolgreich durchgeführt und anschließend beendet.

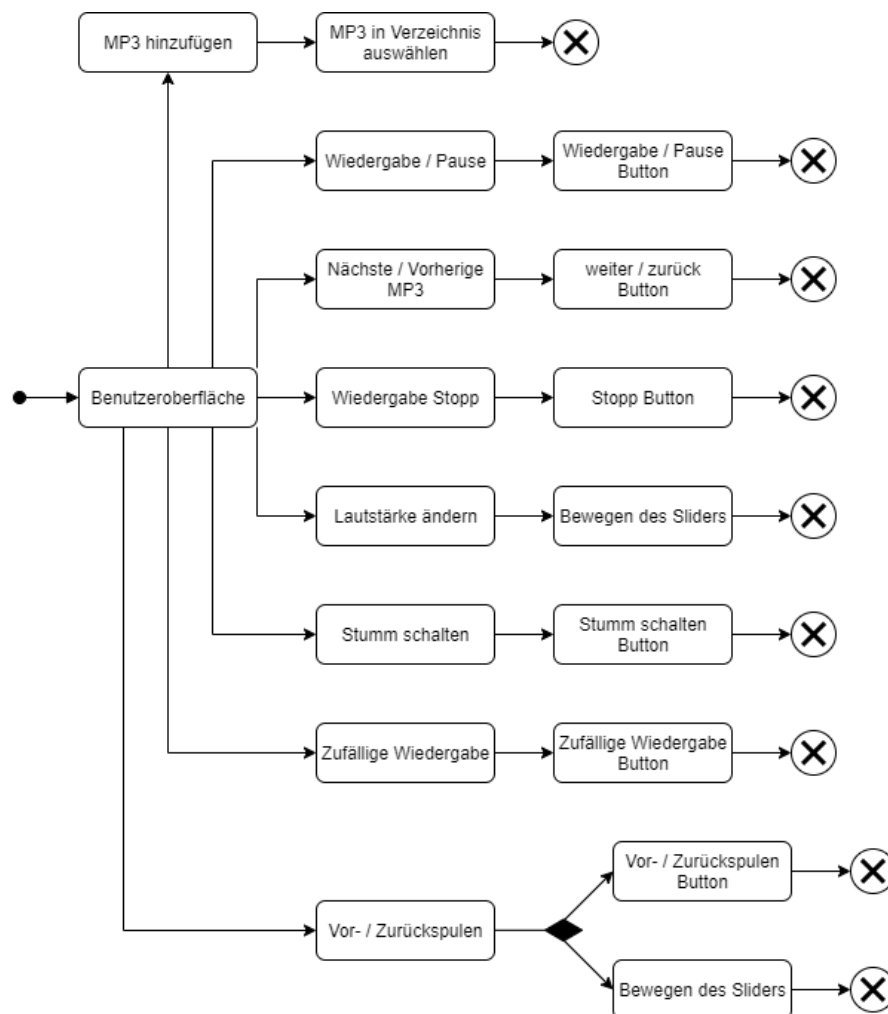


Abbildung 11: Aktivitätsdiagramm

### 4.1.3 Komponentendiagramm

Ein Komponentendiagramm veranschaulicht alle Komponenten, deren Schnittstellen und Abhängigkeitsbeziehungen untereinander, welche verwendet werden. Es zeigt die Entwicklersicht des 4+1 Sichtenmodells.

Es gibt zwei Komponenten, welche miteinander interagieren. Diese sind das Programm und die Datenbank JSON, welche aber nur kommunizieren, wenn MP3 Dateien hinzugefügt oder gelöscht werden.

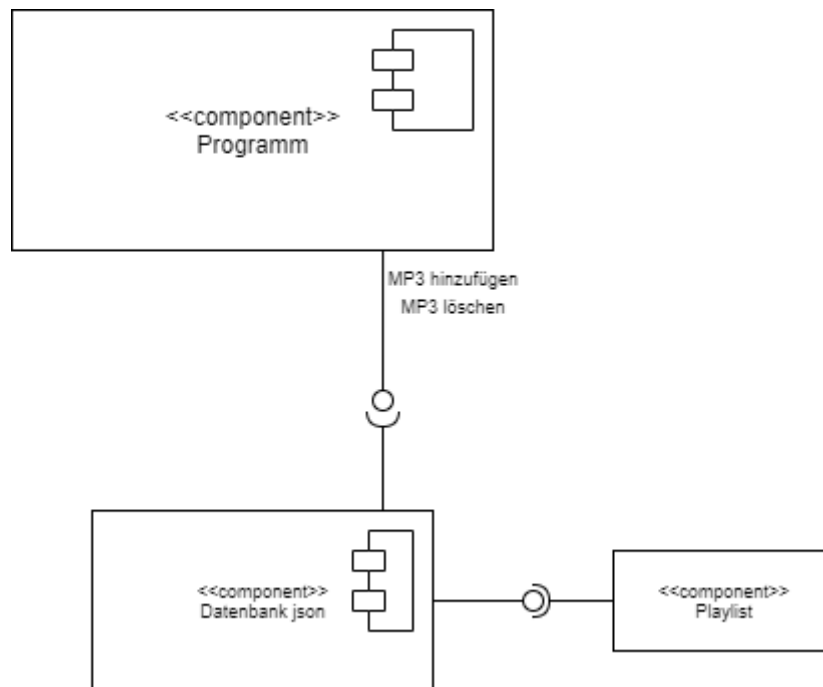


Abbildung 12: Komponentendiagramm

## 4.1.4 Verteilungsdiagramm

Ein Verteilungsdiagramm veranschaulicht die physische Schicht des 4+1 Sichtenmodells.

Das Programm wird im QT Creator ausgeführt, welcher seinerseits von dem Computer des Benutzers ausgeführt wird.

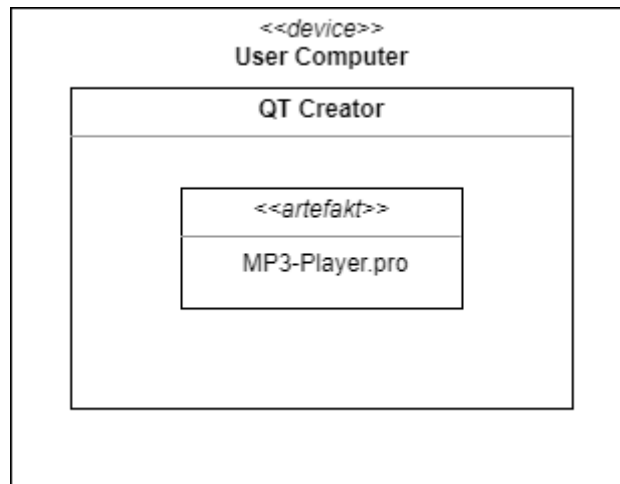


Abbildung 13: Verteilungsdiagramm

## 4.1.5 Anwendungsfalldiagramm

Ein Anwendungsfalldiagramm veranschaulicht das Verhalten der Software in einem Anwendungsfall durch einen Benutzer. Es zeigt die Szenario Sicht des 4+1 Sichtenmodells.

Dargestellt werden die verschiedenen Anwendungen die der Benutzer ausführen kann und welche Wege auf vorherigen aufbauen oder sie erweitern. Im Falle Vor- / Zurückspulen muss logischerweise gerade in diesem Moment eine MP3 abgespielt werden, um die Funktion nutzen zu können.

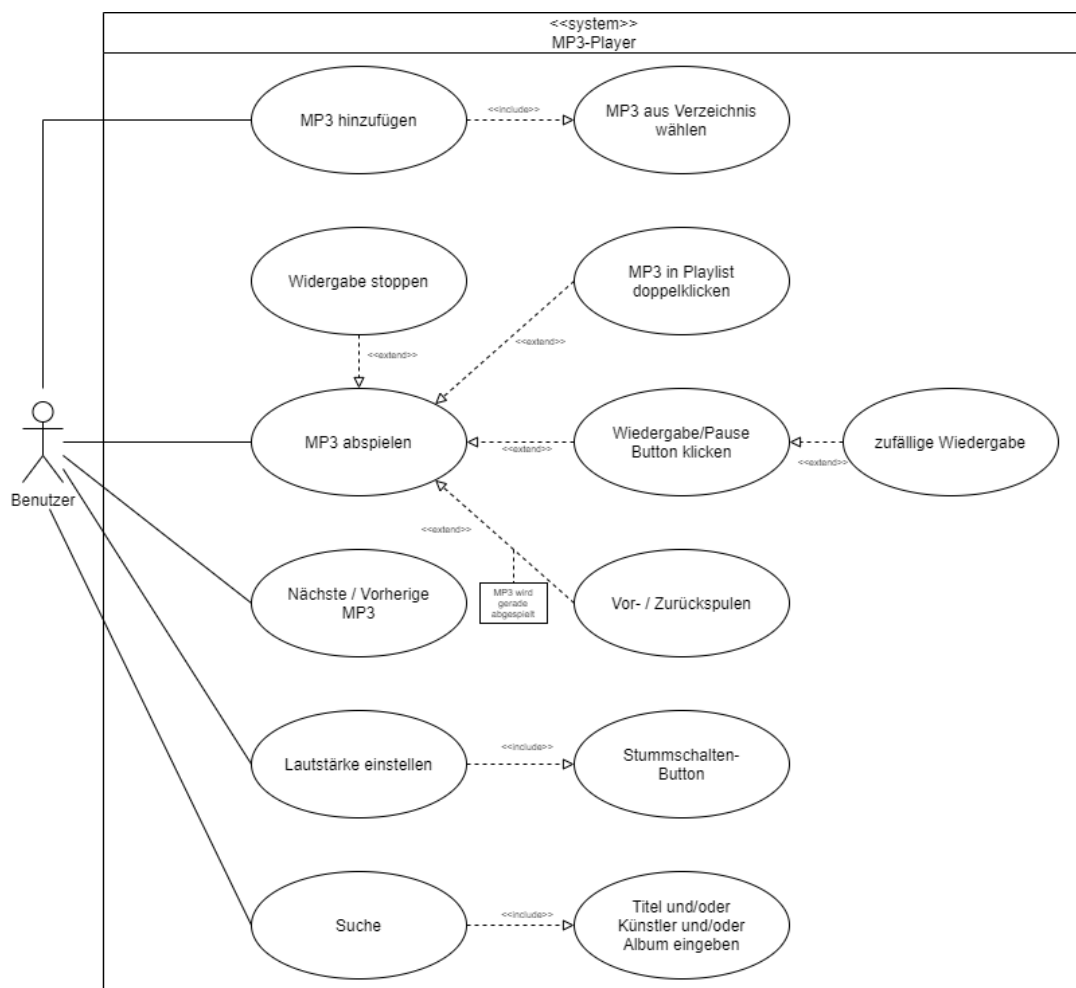


Abbildung 14: Anwendungsfalldiagramm

## 4.2 Datenflussdiagramm

Das Datenflussdiagramm stellt den Informationsfluss innerhalb des Programms dar. Zusehen ist, dass die Playlist die zentrale Einheit des Programms ist. Sie ist mit dem Player verknüpft und übergibt dem Player das aktuell wiederzugebende Lied. Zu der Liste werden die aus dem PC ausgewählten Lieder hinzugefügt. Aus der Json Datei werden die zuvor aus der Playlist gespeicherten Lieder bei einem Neustart geladen. Die Tabelle ist die Darstellungsform der Playlist und die Tabelle teilt der Playlist mit, wenn ein Lied aus der Playlist zu löschen ist.

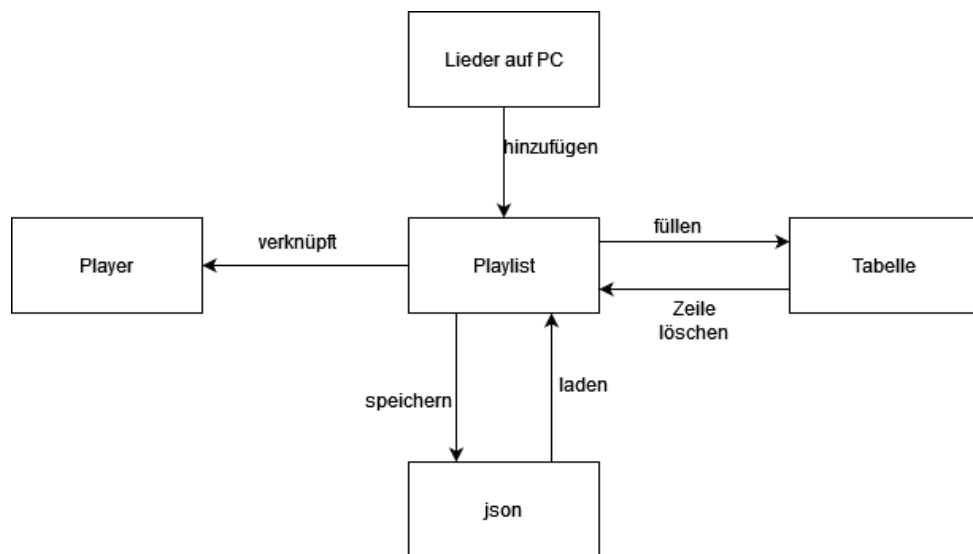


Abbildung 15: Datenflussdiagramm



## 5. Quellen

- Bezüglich Code
  - QT-forum
  - Stackoverflow
  - <https://doc.qt.io/qt-5.12> (29.10 bis 25.11 verwendet)
  - <https://doc.qt.io/qt-5.15> (offizielle QT Dokumentation)