

Produktmuster für das Analyse-Review

DF-1: Einleitung

Ziel ist ein Programm zur Verwaltung und Nutzung von Multimedia-Daten mit Schwerpunkt auf Audio-Dateien (wie z.B. Musikstücke). Das Programm soll plattformunabhängig, und intuitiv zu bedienen sein. Damit das Programm plattformunabhängig auf verschiedenen Betriebssystemen funktioniert, wird JAVA als Programmiersprache benutzt. Als Vorbilder dienen z.B. der Windows Media Player und das Programm WinAmp, die leider stark plattformgebunden sind.

DF-2: Beschreibung des Anwendersystems („A-System“)

DF-2.1: Anforderungen an das A-System:

- Abspielen der Formate .mp3, .ogg, .wav und stream (online Übertragung)
- Bereitstellen vieler Möglichkeiten zur Strukturierung der Datenbestände durch den Benutzer wie Sortieren (hierarchisch, gruppiert, in Listen) und archivieren(speichern und löschen von Dateien)
- Daten sollen in einem einheitlichen Objekt gespeichert werden, damit diese Verwaltung ermöglicht wird
- Dieses Objekt muss alle Daten speichern und auslesen können, die zum Abspielen, Verwalten und Anzeigen notwendig sind
- Bereitstellen einer Schnittstelle die gewährleistet, dass Daten, die in verschiedenen Formaten vorliegen, einheitlich behandelt werden können
- selbstständige Erkennung des Dateiformats (durch den Manager)
- Eingaben des Benutzers erfolgen ausschliesslich über die GUI
- Die GUI stellt Bedienkomfort über eine übersichtliche Struktur, aussagekräftige Symbole/Beschriftungen, die eine intuitive Bedienung zulassen, sicher und ist in zwei Teile aufgeteilt: Wiedergabe und Playlist

DF-2.2: Spezifikation des A-Systems:

- AudioReader stellen die zum Auslesen der Daten in verschiedenen Formaten benötigten Funktionen zur Verfügung
- Ein Manager wählt den zum Format passenden AudioReader und erstellt PlayItems, die dann abgespielt werden können
- PlayItems setzen sich zusammen aus:
 - Location (Dateiname + Pfad/URL)
 - Format (mp3, usw.)
 - ID-Tag-Daten (Titel, Interpret, Album, Genre, Datum, etc.)
- Eine MediaLibrary dient zum Katalogisieren der Mediendaten und stellt eine Liste von PlayItems dar.
 - alle abspielbaren Dateien werden in ihr verwaltet
 - stellt die Daten zur Verfügung und erlaubt Operationen darauf
 - ist durch den Anwender nur beschränkt manipulierbar
 - Einträge lassen sich in eine Playlist übernehmen, sie sind auch hier als PlayItems gespeichert
- PlayLists sind ebenso Listen von PlayItems
 - diese müssen aber nicht zwingend aus der MediaLibrary kommen, sondern können alternativ auch von Wechseldatenträgern stammen (z.B. von einer Audio-CD)
 - die Playlist stellt die Schnittstelle zum Player dar
 - sie ist durch den Anwender durch Datenbankoperationen direkt manipulierbar (Hinzufügen, Löschen, Sortieren, etc)
 - können auch Filteroperationen ausführen
 - PlayLists können importiert und exportiert werden, wobei primär der Austausch von PlayLists im .m3u oder .pls Format, wie sie in WinAmp benutzt werden, angedacht
 - HTML-Export und SQL-Datenbankzugriffe können per Plugin realisiert werden
 - Einträge aus der Playlist werden als PlayItems an den Player übergeben und fungieren als Quelle für den AudioInputStream
- Der Player spielt die Daten aus der Quelle des PlayItems, das aktuell in der Playlist gezeigt wird, ab
 - er kann weiterhin die Playlist anweisen ein anderes PlayItem zu liefern
 - Modi zur PlayItem Wiedergabe (shuffle, loop, etc) können auch über Player verändert werden
 - Operationen wie Suchlauf, Stop und Pause können vom Player direkt auf aktiven PlayItems ausgeführt werden
- Die GUI dient als Eingabe-Schnittstelle durch Bedienelemente, sowie als Ausgabe für etwaige Fehlermeldungen, Hinweise o.Ä.
- Geeignete Kommunikation zwischen den Elementen wird durch Exceptions, Events und öffentlichen Methoden gewährleistet

Des Weiteren wird die Software so ausgelegt sein, dass Antwortzeiten für Suchanfragen nicht länger als fünf Sekunden betragen sollen, und dass mindestens 1000 Datenbankeinträge, also Dateien verwaltet werden können.

DF-2.2.1: Einordnung des A-Systems

- System zum Anlegen, Verwalten und Speichern einer Mediendatenbank mit Wiedergabefunktion für die Gespeicherten Daten über eine GUI, welche die Schnittstelle zwischen Anwender und Programm darstellt.
 - Die GUI wird vom Benutzer über die Tastatur und die Maus bedient.
 - Die Funktionen zur Wiedergabe sind quellenabhängig (Streams vs. Dateien; Internet Radio kann man z.B. nicht zurückspulen)
- ### DF-2.2.2: Schnittstellen des A-Systems
- Bereitstellen von Player-Funktionen:
 - aktuelles PlayItem aus einer PlayList zurückgeben
 - nächstes und voriges PlayItem zurückgeben
 - Bereitstellen von PlayList-Funktionen:
 - PlayItems zur PlayList/Media Library hinzufügen/entfernen
 - PlayItems in PlayList/MediaLibrary suchen und zurückgeben
 - Auswahl von PlayItems in gewünschter Form aus der PlayList/MediaLibrary zurückgeben
 - Input: Sound-Device input (aus Datei oder Stream)
 - Output: Sound-Device output und Runtime Informationen
 - Datenaustausch mit der GUI
 - Entgegennahme von Player Kommandos (über Eingabefelder der GUI)
 - Rückgabe von Fehlermeldungen
 - Zugriff auf Filesystem:
 - Erfassung von Dateien in der MediaLibrary
 - Sequentielles Lesen (Stream) und relatives Lesen (Datei) von Mediendaten und Bereitstellung als PlayItem
 - Lesen und schreiben der PlayList
 - Schnittstelle für das einbinden neuer Formate zum Lesen (AudioReader) und schreiben (Writer)

DF-2.3: Konstruktion des A-Systems

DF-2.3.1: Beschreibung des organisatorischen Systems (O-System)

Das A-System stellt den Player und die GUI Funktionen Bereit

DF-2.3.1.1: Aufbauorganisation

- MediaLibrary und PlayList existieren immer parallel, unterscheiden sich aber in einigen Punkten
- PlayItems sind als Einträge in einer dieser Listen zu verstehen, die Informationen über die Medienquelle, den Künstler etc. bereitstellen
- Der Player spielt PlayItems aus der PlayList ab, welche in der MediaLibrary gespeichert werden können

DF-2.3.1.2: Ablauforganisation

DF-2.3.2: Beschreibung des technischen Systems (t-System)

DF-2.3.2.1: SW-System:

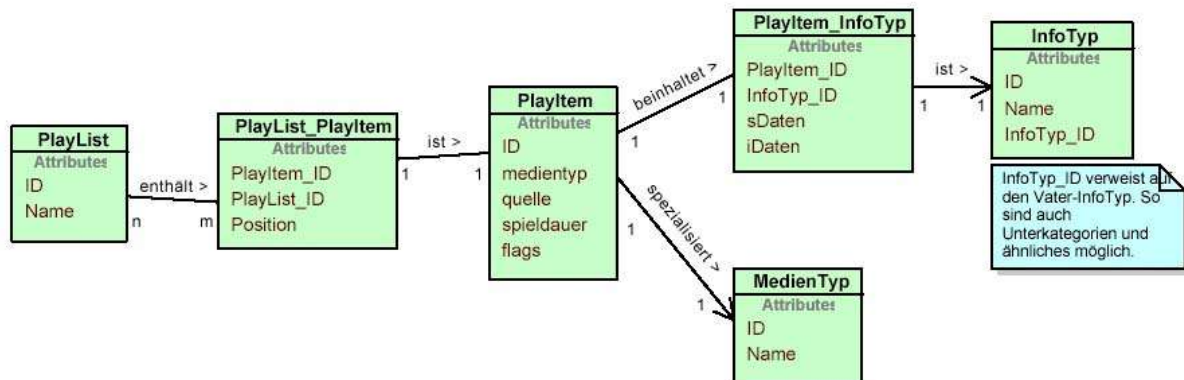
- Das SW-System des Ziel-t-Systems muss über eine ordnungsgemäß funktionierende Java-VM verfügen, da Java als Programmiersprache benutzt wird. Dies kann jedoch durch den Gebrauch von nativen Compilern umgangen werden
- Das Programm wird dann als normale Applikation laufen

DF-2.3.2.2: Basis-System: (HW, Basis-SW)

- Die Basis-SW muss Java unterstützen
- Für die HW ist ein Sound-Device zwingend notwendig
- des weiteren muss genügend Speicherplatz zur Verfügung gestellt werden, damit die MediaLibrary ordentlich gespeichert werden kann.

DF-2.3.2.3: Sonstige technische Einrichtungen

DF-3: Datenmodell



DF-4: Funktions- und Ablaufmodell

DF-4.1 Übersicht über die Funktionen:

- Datenbankverwaltung für Multi-Media-Dateien
 - Hinzufügen, Ersetzen, Löschen, Bearbeiten der Einträge
 - Sortieren, Filtern und Suchen von Einträgen
 - Laden, Speichern, Import/Export der Playliste
- Playlistfunktionen
 - Bewegen in der Playlist
 - Bestimmte Modi für die Reihenfolge
- Player für die Multi-Media-Dateien
 - Abspielen
- Funktionen der GUI
 - Wiedergabefenster
 - AbspielButtons (Play, Stop, Pause, Vor-/Rücklauf, nächstes, vorheriges, Shuffle, Repeat)
 - Fortschrittsanzeige des aktuellen Liedes
 - Anzeige: Titel, Interpret, Länge
 - Optional: Anzeige von Mono/Stereo, Kbps
 - Lautstärkeregler
 - Playlistfenster
 - Liste der Playltems
 - Markierung des aktuellen Liedes
 - Möglichkeit des Drag'n'Drop
 - Buttons für Laden/Speichern
 - Optional: Sortieren

DF-4.1.1 Use Cases:

Titel: Ansichtsoption, kurz Ans_op

Zweck:

Ändern der Ansicht

Involvierte Aktoren:

Benutzer

Auslösende Ereignisse:

Der Benutzer wählt den Menüpunkt „Ansicht anpassen“

Für die Auslösung benötigte Daten:

Aktuelle Konfigurationsdaten des Programms

Ausnahme und Fehlersituation:

Nicht kompatible Skin wird geladen alte Konfiguration beibehalten

Teilschritte und Aktionen:

Der Benutzer wählt die gewünschte Ansicht. Diese wird dann angezeigt (in Form einer anderen Skin, oder in der Art der Darstellungsweise der PlayItems in der PlayList

Ergebnisse und Nachbedingungen:

Ansicht bzw. Skin wurde geändert; Programm kehrt zum vorherigen Zustand zurück.

Titel: Editieren von Bibliotheksdaten kurz Bib_ed

Zweck:

Informationen zu vorhanden Musikdateien eingeben und/oder ändern (Name des Songs, Interpret/Band, Länge des Songs,...)

Involvierter Aktoren:

Benutzer

Auslösende Ereignisse:

Der Benutzer markiert eine Datei und wählt den Button "Infos neu/bearbeiten"

Für die Auslösung benötigte Daten:

Bereits in der Bibliothek vorhandene Daten

Ausnahme- und Fehlersituationen:

Speicherplatz ist voll, falsche Eingabe Fehlermeldung

Teilschritte und Aktionen:

Die Daten werden nach und nach in die zu Verfügung stehenden Felder eingetragen. Bei Bedarf können noch weitere Felder eingefügt werden. Wenn alle gewünschten Felder ausgefüllt wurden, wird der Button "speichern" zur Sicherung der Daten gedrückt, dadurch wird das Fenster geschlossen und der Benutzer kehrt zu der Bibliothek zurück.

Ergebnisse und Nachbedingungen:

Die eingegebenen Daten werden in der dazugehörigen Datei gespeichert. Die Eingabemaske muss auf "null" zurückgesetzt werden. Bei Aufruf der selben Datei müssen die eingegebenen Daten wieder angezeigt werden.

Titel: Hinzufügen zu Bibliothek, **kurz:** Bib_hinzu

Zweck:

Hinzufügen einer Mediadatei zu einer vorhandenen Bibliothek

Involvierte Akteure:

Benutzer

Auslösende Ereignisse:

Der Benutzer betätigt „(Datei) hinzufügen“; gibt eine Datei an die er zu einer vorhandenen Bibliothek hinzufügen will

Für die Auslösung benötigte Daten:

Pfad der Datei, Dateiname, Bibliothek (Name und Ort), evtl. Leseberechtigung für die Datei

Ausnahme- und Fehlersituation:

Datei nicht vorhanden Fehlermeldung

Datei schon in Bibliothek vorhanden Dialog „Überschreiben?“

Keine Schreib-/Leseberechtigung Fehlermeldung

Teilschritte und Aktionen:

Der Benutzer nennt eine Datei, die er zu einer vorhandenen Bibliothek hinzufügen will. Die Daten der Datei werden dann in der Bibliothek gespeichert. Anschließend wird die Datei ausgelesen um ggf. ID-Tag oder andere Informationen angezeigt werden können.

Ergebnisse und Nachbedingungen:

Ein Verweis auf die Datei ist in der Bibliothek gespeichert.

Titel: Lautstärkeregelung, **kurz** Vol_mod

Zweck:

Die Lautstärke ändern

Involvierte Akteure:

Benutzer

Auslösende Ereignisse:

Der Lautstärkereglern wird verändert/bewegt

Für die Auslösung benötigte Daten:

Aktuelle Konfigurationsdaten des Programms

Ausnahme- und Fehlersituationen:

Lautstärkemodifizierung schlägt fehl Beibehaltung der bisherigen Lautstärke

Teilschritte und Aktionen:

Der Regler wird bewegt; AudioOutputStream an das Sound Device wird entsprechend modifiziert

Ergebnisse und Nachbedingung:

Lautstärke wurde gemäß den Benutzereingaben modifiziert.

Titel: Löschen eines Eintrages aus der Bibliothek, **kurz** Bib_Item_del

Zweck:

Entfernen eines Eintrages aus der Bibliothek

Involvierte Akteure:

Benutzer

Auslösende Ereignisse:

Der Benutzer markiert den Eintrag und gibt den Löschbefehl.

Für die Auslösung benötigte Daten:

Dateiquelle des zu löschenden Elements

Ausnahme- und Fehlersituationen:

Der Eintrag, der gelöscht werden soll, ist nicht mehr vorhanden
Fehlermeldung

Teilschritte und Aktionen:

Markieren und „löschen“ drücken.

Ergebnisse und Nachbedingungen:

Der Eintrag wurde gelöscht. Fokus wird auf den nächsten Eintrag der Liste gesetzt.

Titel: Neue ID-Tags Eintragen, **kurz** ID_new

Zweck:

Vorhandene Audiodateien werden mit ID-Tag-Daten ergänzt.

Involvierte Akteure:

Benutzer.

Auslösende Ereignisse:

Benutzer übt Rechtsklick der Maus über Liedauswahl aus, oder benutzt dafür bereitgestellten Button/Menüeintrag.

Für die Auslösung benötigte Daten:

Daten einer Audiodatei.

Ausnahme- und Fehlersituationen:

Unzulässige Dateneingabe, Audiodatei kann nicht mit ID-Tag-Daten versehen werden, wenn das Format es nicht zulässt, Filesystem I/O Fehler (Zugriffrechte, Datei nicht mehr vorhanden) Fehlermeldung

Teilschritte und Aktionen:

Dem Benutzer wird eine Eingabemaske mit den möglichen Eingabefeldern bereitgestellt. Gültigkeit bestimmter Eingabewerte können durch Bereitstellung von „List-Boxes“ sichergestellt werden. Nach Eingabe der Daten kann der Benutzer die (auch unvollständigen) Daten der Audiodatei befügen, oder verwerfen.

Ergebnisse und Nachbedingungen:

Die gewählte Audiodatei wurde um ID-Tag-Daten ergänzt. Die ergänzte Datei wird automatisch neu in die Media-Library eingelesen.

Titel: Modus ändern, kurz Mod_ed

Zweck:

Ändern des Abspielmodus (Repeat (one/all), Shuffle)

Involvierte Akteure:

Benutzer

Auslösende Ereignisse:

Der Benutzer aktiviert den entsprechenden Button per Maus

Für die Auslösung benötigte Daten:

PlayList und der aktuelle Abspielmodus

Ausnahme- und Fehlersituationen:

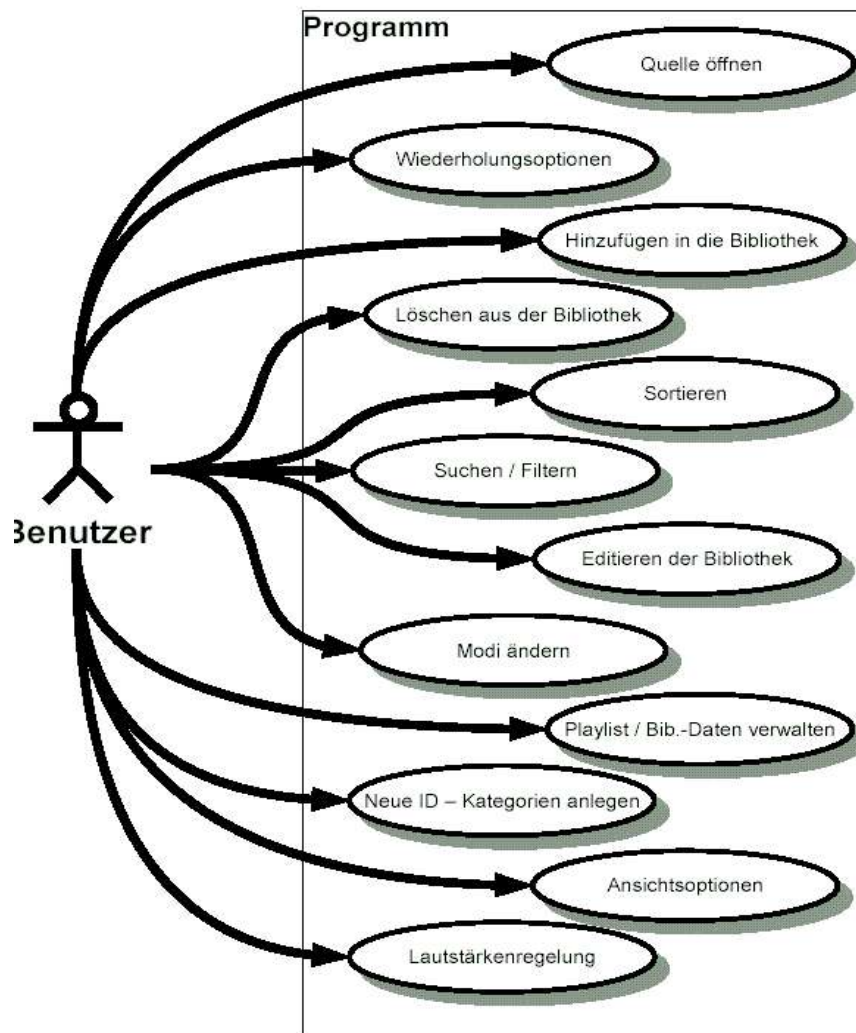
keine

Teilschritte und Aktionen:

Repeatbutton bzw. Shufflebutton anklicken

Ergebnisse und Nachbedingungen:

Löst der Benutzer den Repeatbutton bzw. den Shufflebutton aus, wechselt der Repeatmodus bzw. Shufflemodus zwischen „an“ und „aus“. Ist der Repeatmodus an, stoppt der Player nicht nach dem letzten Lied der PlayList, sondern fängt wieder mit dem ersten an, andernfalls stoppt die Wiedergabe. Ist der Shufflemodus an, werden die einzelnen Lieder der PlayList in zufälliger Reihenfolge wiedergegeben, andernfalls der Reihe nach. Ausnahme ist der "Repeat one" Modus, in dem nur ein Track wiederholt wird.



Titel: PlayList und Bibliotheksdaten verwalten, **kurz** PL_Bib_Verw

Zweck:

Playltems aus der PlayList aus der MediaLibrary, oder ganze PlayListen laden oder speichern

Involvierte Akteure:

Benutzer

Auslösende Ereignisse:

Mittels Mausklick auf entsprechend beschriftete Buttons oder durch Drag & Drop einzelner Liedeinträge in der Liste einfügen/entfernen.

Ganze PlayListen per Buttonklick im-/ oder exportieren bzw. laden und speichern.

Für die Auslösung benötigte Daten:

Für das Hinzufügen werden vorhandene Sounddateien in den unterstützten Formaten benötigt und/oder die Medienbibliothek.

Das Löschen der Einträge, das Sortieren der Liste sowie das Exportieren einer Liste benötigt die Einträge der Liste selbst.

Import einer PlayList setzt das Vorhandensein einer solchen Datei voraus.

Ausnahmen und Fehlersituationen:

Zu ladende Datei nicht vorhanden, I/O-Fehler, falsches Format
Fehlermeldung

Zu speichernde Datei schon vorhanden Dialog „Überschreiben?“

Teilschritte und Aktionen

Hinzufügen von Einträgen in die Liste

Löschen von Einträgen aus der Liste

Importieren vorhandener PlayListen

Exportieren neuer PlayListen

Sortieren der aktuellen PlayListItems

Manuelles Umsortieren der PlayItem-folge

Ergebnisse und Nachbedingungen:

Anzeige einer gewünschten Playltem-folge in einer PlayList „Verwaltung“ der einzelnen PlayListen , um zukünftige Listenerstellungen zu vereinfachen.

Titel: Quelle öffnen, **kurz** File_open

Zweck:

Eine Datei öffnen, um sie abzuspielen oder zu verändern

Involvierte Akteure:

Benutzer

Auslösendes Ereignis:

Betätigen des Buttons/der Schaltfläche „hinzufügen“ oder „Datei öffnen“

Für die Auslösung benötigte Daten:

Pfad der zu öffnenden Datei; Lese-/Schreiberechtigungen

Ausnahme- und Fehlersituationen:

Falsches Dateiformat, keine Berechtigung Fehlermeldung

Teilschritte und Aktionen:

Dialog einblenden, Benutzer Datei auswählen lassen

Ergebnisse und Nachbedingungen:

Auf die Datei kann nun zugegriffen werden, und ist in der PlayList als PlayItem vorhanden.

Titel: Sortieren, **kurz** Sort

Zweck:

Sortieren der Datenbank nach einem bestimmten Kriterium (nach Titel, Interpret, Genre usw...)

Involvierte Akteure:

Benutzer

Auslösende Ereignisse:

Der Benutzer betätigt den entsprechenden Button per Maus

Für die Auslösung benötigte Daten:

Datenbank mit Einträgen. Sortier-Kriterium.

Ausnahme- und Fehlersituationen:

Datenbank ist leer keine Aktion

Teilschritte und Aktionen:

Sortierbutton und entsprechendes Sortier-Kriterium anklicken.

Ergebnisse und Nachbedingungen:

Der Benutzer klickt den Sortier-Button, die GUI sendet an die MediaLibrary einen Befehl, die DB zu sortieren und das Ergebnis zurück zu liefern. Kommt das Ergebnis bei der MediaLibrary an, gibt diese es weiter an die GUI, welche das Ergebnis dann schließlich wieder zum Benutzer zurück gibt. Als Kriterien dienen die ID-Tags.

Titel: Suchen/Filtern, **kurz:** Su_Fi

Zweck:

(Aus-)suchen von Dateien anhand von bestimmten Kriterien

Involvierte Akteure:

Benutzer

Auslösendes Ereignis:

Klick auf Button „Suchen/Finden“

Für die Auslösung benötigte Daten:

Zu durchsuchende Playlist bzw. MediaLibrary, ausgefüllte ID-Tags und die Suchkriterien

Ausnahme- und Fehlersituationen:

Keine oder leere Playlist oder MediaLibrary, nicht ausgefüllte ID-Tags

Fehlermeldung

Keine passende Datei gefunden Dialogbox

Teilschritte und Aktionen:

Kriterien des Benutzers abfragen, anwenden auf Playlist oder MediaLibrary, entsprechende Darstellung der Suchergebnisse.

Ergebnisse und Nachbedingungen:

Ansicht der Suchergebnisse.

Titel: Wiedergabe pausieren, **kurz:** Pause

Zweck:

Anhalten der Wiedergabe der Datei oder des Streams

Involvierte Akteure:

Benutzer

Auslösendes Ereignis:

Der Benutzer drückt den „Pause“-Button

Für die Auslösung benötigte Daten:

Quelle des PlayItems

Ausnahme- und Fehlersituationen:

Keine Wiedergabe keine Aktion

Teilschritte und Aktionen:

Die aktuelle Position in der Datei wird gespeichert, sodass die Wiedergabe ab dieser Position nach der erneuten Drücken auf „Pause“ oder „Play“ ermöglicht wird.

Ergebnisse und Nachbedingungen:

Kein AudioStream an das Sound Device, Player im Wartezustand.

Titel: Wiedergabe starten, **kurz:** Play

Zweck:

Abspielen von einer Datei oder Stream

Involvierte Aktoren:

Benutzer

Auslösendes Ereignis:

Der Benutzer drückt den „Play“-Button

Für die Auslösung benötigte Daten:

Ein PlayItem.

Ausnahme- und Fehlersituationen:

Das Format des abzuspielenden Musikstücks entspricht nicht den vorgesehenen Formaten Fehlermeldung, Datei oder Stream wird nicht abgespielt

Die Zieldatei wurde nicht gefunden (verschoben, gelöscht)

Fehlermeldung, Datei oder Stream wird nicht abgespielt

Teilschritte und Aktionen:

Nach dem Drücken des „Play“-Buttons wird ein „PlayItem“ aus der „PlayList“ geladen, der Player sucht anschließend den passenden Dekoder aus und spielt das Musikstück ab.

Ergebnisse und Nachbedingungen:

AudioOutputStream an das Sound Device

Titel: Wiedergabe stoppen, **kurz:** Stop

Zweck:

Abbruch der Wiedergabe der Datei oder des Streams

Involvierte Aktoren:

Benutzer

Auslösender Ereignis:

Der Benutzer drückt den Knopf „Stop“

Für die Auslösung benötigte Daten:

keine

Ausnahme- und Fehlersituationen:

Es läuft keine Wiedergabe keine Aktion

Teilschritte und Aktionen:

Nach betätigen des „Stop“-Buttons wird der AudioOutputStream abgebrochen

Ergebnisse und Nachbedingungen:

Kein AudioStream an das Sound Device

DF-4.2: Funktionsbeschreibungen

DF-4.2.1: Benutzerfunktionen:

- Datenbankverwaltung
 - Der Benutzer kann die Datenbank um weitere Einträge erweitern, vorhandene löschen oder durch neue Einträge ersetzen. Weiter kann er Einträge bearbeiten, in dem er z.B. den Interpreten oder den Titel der Datei ändert.
 - Der Benutzer kann nach eigenen Kategorien seine Datenbank sortieren und Elemente herausfiltern, denkbar sind z.B. Lieblingsinterpreten. Darüber hinaus gibt es eine Suchfunktion nach den Eigenschaften der Einträge.
 - Der Benutzer kann selbst zusammengestellte PlayLists laden, speichern, importieren und exportieren.
- Playlistfunktionen
 - Der Benutzer kann zwischen normaler Reihenfolge, Shufflefunktion und Repeatfunktion wählen. Außerdem kann zum nächsten Eintrag oder zum vorherigen zurückgegangen werden oder am aktuellen Titel verharret werden.
- Playerfunktionen
 - Dem Benutzer wird die Möglichkeit gegeben, eine Datei abzuspielen, das Abspielen zu unterbrechen, zu stoppen, in der Datei nach vorne bzw. zurückzugehen. Die Basis des ganzen bildet das Öffnen der Datei.
- Funktionen der GUI
 - Im Wiedergabefenster soll der Benutzer die bekannten Abspielbuttons finden. Weiter wird ihm eine Lautstärkeregel zur Verfügung gestellt.
 - Im Playlistfenster wird dem Benutzer eine Möglichkeit zum manuellen Sortieren der Einträge gegeben, eventuell auch über Buttons. Er kann über Buttons Dateien der Liste hinzufügen oder sie löschen, eventuell auch Speichern und Laden von Playlisten über Buttons.
 - Beide Fenster können über Buttons minimiert, ausgeblendet oder geschlossen werden.

DF-4.2.2: Dienstleistungsfunktionen:

- Funktionen der GUI
 - Dem Anwender im Wiedergabefenster Informationen über den Titel, den Interpreten, die Länge und den Fortschritt des aktuellen Liedes angezeigt.
 - Im Playlistfenster sieht der Benutzer die Liste der ausgewählten Lieder, wobei der aktuelle Titel markiert ist.
- Playlistfunktionen
 - Automatisch wird das nächste Stück angespielt, die richtigen Codecs werden gewählt, ID-Tags werden eingelesen und die PlayList gespeichert.

DF-4.2.3: Betriebsorientierte Funktionen:

- Playerfunktionen
 - Der Player muss mit der GUI und der Datenbank kommunizieren.

DF-4.3: Ablaufmodell

DF-4.3.1: Benutzermodelle

DF-4.3.2: Arbeitsabläufe, Dialoggestaltung

- Der Player benötigt Quelle und Quellentyp der Datei, die gespielt werden soll, um einen Funktionsaufruf für die unter 4.2.1 beschriebenen Funktionen. Des weiteren gibt der Player Dateiinformatoren an die GUI weiter.

DF-4.3.3: Masken- und Listenformate

DF-4.4: Sonstige Schnittstellen des Software-Systems

DF-4.4.1: Hardware-Schnittstellen:

- Sound-Device

DF-4.4.2: Schnittstellen zur Basissoftware:

- Schnittstellen zum Basissystem stellen viele der Javaklassen aus der Standardbibliothek dar, z.B. für Datei- und Netzwerkzugriff. Ebenso wird die Sound-API benötigt, um die Audiodaten auszugeben.

DF-4.4.3: Schnittstellen zu Nachbarsystemen:

- Codecs von Drittanbietern
- Import/Export der PlayList

DF-4.5: Nicht-funktionale Anforderungen

DF-4.5.1: Leistungsanforderungen:

- Die Such-/Sortier-/Filterfunktionen müssen effizient arbeiten
- Niedrige Systemanforderungen
- Die Daten müssen unter normaler Systemlast ruckelfrei abgespielt werden können.

DF-4.5.2: Ausfallsicherung:

- Abfangen von Exceptions
- Codesicherheit (auf ArrayOutOfBounds, Stack-Overflow, Buffer-Overflow, Signed-Variables überprüfen)
- Behandlung kritischer Situationen
- Abspielen einer leeren Playlist
- Quelle aus PlayItem nicht mehr vorhanden
 - Bufferunderrun
- Dabei ist eine Betrachtung der in der Java-API schon integrierten Fehlerbehandlung zu berücksichtigen.

DF-4.5.3: Vertraulichkeit:

- Entfällt

DF-4.5.4: sonstige Qualitätsanforderungen:

- Die grafische Benutzeroberfläche sollte übersichtlich gestaltet werden und intuitiv zu bedienen sein. Das bedeutet eine schnell erfassbare Anordnung der Buttons und Anzeigeelemente ebenso wie klare Kontraste und gute Lesbarkeit angezeigter Texte. Optische ansprechende Skins wären dabei zu bevorzugen.

DF-5: Test- und Einführungsbedingungen

Entfallen da es bei dem Projekt nicht darum geht ein anderes Programm abzulösen

DF-5.4: Ausbaumöglichkeiten:

- Für jeden selbst, da open-source
- Erweiterte Version nach Aufgabenstellung

DF-6: Abnahme-Testentwurf

DF-6.1: Test-Anforderungen

- Der Test soll das Programm unter möglichst vielen verschiedenen Voraussetzungen laufen lassen, d.h. beim Test sollen sich folgende Komponenten ändern:
 - Hardware
 - Betriebssystem
 - Systemlasten
 - Playlisten
 - Belastung

DF-6.2: Testkonfiguration und -vorgehensweise

- Abarbeiten der Use Cases.

DF-7: Anhang

DF-7.1: Datenmodell:

