**Module Interface Specification For The Library Module**

Brian Alexander

# Introduction

The library form provides a graphical user interface for control and display of library features.

# Requirements

## Services Provided By this Module to Its Users

|  |  |  |
| --- | --- | --- |
| Service | Provided By | Tested By |
| 1. Stores file paths and media information that can be sorted, filtered, and searched. | getArtistList, getSongList |  |

## Requirements for Services Needed By The Module

The playlist form relies on the playlist module as a target for messages.

# Exceptions

None.

# Access Method Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Exceptions: None** | | | | | |
| **Access Methods** | **Parameter name** | **Parameter type** | **Parameter Info** | **Performance**  **Best - Worst** | **–Mapping to Services Provided** |
| getArtistList | f | LibraryFilter | Specifies the filter to apply to the artist list |  | 1 |
| getSongList | f | LibraryFilter | Specifies the filter to apply to the song list |  | 1 |

# Local Dictionary

## Parameter Terms

|  |  |
| --- | --- |
| **Parameter Term** | **Meaning** |
|  |  |
|  |  |

## Local Terms

|  |  |
| --- | --- |
| **Term** | **Meaning** |
|  |  |
|  |  |

## Local Conditions

|  |  |
| --- | --- |
| **Term** | **Meaning** |
|  |  |
|  |  |

## Local Data Types

|  |  |
| --- | --- |
| **Type** | **Value Space** |
| LibraryFilter | Should contain strings that are used to filter songs using metadata. |

# Access Methods Effects

getArtistList will filter every song in the library using the filter specified. It will return a list of artists associated with the filtered songs.

getSongList will filter every song in the library using the filter specified. It will return a list of song titles associated with the filtered songs.

# Events Signaled

None

# Externally Visible States of the Module

None

# Module Configuration Parameters

None.

# Interface Design Issues

None.

# Implementation Notes.

# Efficiency Guide

# Test Cases

# Questions for Reviewers

Reviewers of this module interface specification should answer the following questions as they think about the specification.

## Requirements Validity

1. For each service provided by the module, is the service valid for all expected uses of this module? If not, give an example of a use where the service is not valid. (Systems Engineer)
2. For each service provided by the module, is the service valid for all expected configurations and versions of this module? If not, give an example of a needed configuration or version where the service is not valid. (Systems Engineer)
3. For each service needed described in this specification, is a module (or set of modules) identified that this module is allowed to use to satisfy the need? (Architect, Tester)
4. Are there cases where the interface specification could not be satisfied or was incomplete+? If so, how should it be changed? (Architect)

## Requirements Sufficiency

1. Does the set of services provided specify all of the services that will be needed by users of this module? Are there any services defined that are not identified in the requirements? (Systems Engineer)
2. Does the set of services needed specify all of the services that this module will need from other modules in order to operate correctly? What services are needed that are not identified in the requirements? (Architect)

## Consistency Between Services Provided and Access Programs

1. For each Services Provided described in this specification, which access program(s) can be used to satisfy the service? (Architect , Developer, Tester)
2. For each access program and signal specified in sections and which Service Provided is satisfied by the access programs? (Architect , Developer, Tester)

## Access Program Adequacy

1. Is the set of access programs and signals, including exceptions, sufficient to satisfy the uses needs of modules that are allowed to use this module? (Architect)
2. Are there access programs that should be combined into one access program? (Developer)
3. Are there single access programs that should be refactored into several different access programs? (Developer)
4. Are the performance requirements adequate for the uses that will be made of this module? (Tester)

## Implementation of Variability

1. Which variability (or variabilities) does this module implement? (Systems Engineer)
2. Are all values in the range defined in the parameters of variation accounted for in implementing each variability? Which values are not? (Developer)
3. Can the variabilities be bound at the time specified in the commonality analysis for the variabilities? (Architect)