**Use Case 1: UserClicksTogglePlaybackButton**

**Overview:**

Allows user to pause or resume music playback.

**Preconditions:**

1. The jMPD is listening on the configured port.
2. The database is accessible.
3. The jMPC is connected to jMPD on the same configured port.

**Scenario:**

| **Action** | **Software Reaction** |
| --- | --- |
| 1. User clicks playback toggle button | 1a. Client sends "toggle" command to jMPD over port.  1b. Server pauses/resumes playback and sends response to client.  1c. Client toggles button icon. |

**Scenario Notes:**

If the jMPD detects that there are no tracks in the play queue, the toggle command has no effect. The client's toggle playback button icon shall always display the opposite of the player state.

**Post Conditions:**

1. If the client was displaying the "play" icon (daemon paused), then it is now displaying the "pause" icon.
2. If the client was displaying the "pause" icon (daemon playing), then it is now displaying the "play" icon.
3. If the daemon was playing audio, it has now paused playback.
4. If the daemon was in the paused state, it is now playing audio starting from the same position where it was paused.

**Exceptions:**

1. The database cannot be accessed
2. The daemon is not running
3. The client is not connected to the same port as jMPD.

**Required GUI:**

1. jMPC\_Main\_View

**Use Cases Utilized:**

None

**Timing Constraints:**

None

**Use Case 2: UserClicksNextTrackButton**

**Overview:**

Allows the user to skip forward to the next track in the play queue.

**Preconditions:**

1. The jMPD is listening on the configured port.
2. The database is accessible.
3. The jMPC is connected to jMPD on the same configured port.

**Scenario:**

| **Action** | **Software Reaction** |
| --- | --- |
| 1. User clicks next button | 1a. Client sends "next" command to jMPD over port.  1b. Server starts playback of the next track in the playqueue.  1c. Client updates view with current track info. |

**Scenario Notes:**

If the jMPD detects that there are no tracks in the play queue, the next command has no effect. If there is no next track, jMPD will wrap the queue and play the first song.

**Post Conditions:**

1. The daemon is now playing the next song in the queue.
2. The client is now displaying the new song information.

**Exceptions:**

1. The database cannot be accessed
2. The daemon is not running
3. The client is not connected to the same port as jMPD.

**Required GUI:**

1. jMPC\_Main\_View

**Use Cases Utilized:**

None

**Timing Constraints:**

None

**Use Case 3: UserClicksPreviousTrackButton**

**Overview:**

Allows the user to skip back to the previous track in the play queue.

**Preconditions:**

1. The jMPD is listening on the configured port.
2. The database is accessible.
3. The jMPC is connected to jMPD on the same configured port.

**Scenario:**

| **Action** | **Software Reaction** |
| --- | --- |
| 1. User clicks previous button | 1a. Client sends "prev" command to jMPD over port.  1b. Server starts playback of the previous track in the playqueue.  1c. Client updates view with current track info. |

**Scenario Notes:**

If the jMPD detects that there are no tracks in the play queue, the prev command has no effect. If there is no previous track, jMPD will wrap the queue and play the last song.

**Post Conditions:**

1. The daemon is now playing the previous song in the queue.
2. The client is now displaying the new song information.

**Exceptions:**

1. The database cannot be accessed
2. The daemon is not running
3. The client is not connected to the same port as jMPD.

**Required GUI:**

1. jMPC\_Main\_View

**Use Cases Utilized:**

None

**Timing Constraints:**

None

**Use Case 4: UserClicksStopButton**

**Overview:**

Allows user to end playback and leave the daemon paused at the beginning of the current song.

**Preconditions:**

1. The server is listening on the configured port.
2. The database is accessible.
3. The jMPC is connected to jMPD on the same configured port.

**Scenario:**

| **Action** | **Software Reaction** |
| --- | --- |
| 1. User clicks stop button | 1a. Client sends "stop" command to jMPD over port.  1b. Server stops music playback.  1c. Client updates progress bar in the view to beginning. |

**Scenario Notes:**

If the jMPD detects that there are no tracks in the play queue, the stop command has no effect.

**Post Conditions:**

1. The daemon is now paused at the beginning of the current track in the queue.

**Exceptions:**

1. The database cannot be accessed
2. The daemon is not running
3. The client is not connected to the same port as jMPD.

**Required GUI:**

1. jMPC\_Main\_View

**Use Cases Utilized:**

None

**Timing Constraints:**

None

**Use Case 5: UserUpdatesDatabase**

**Overview:**

Allows user to manually trigger an update of the jMPD database through the client GUI. Effectively lets the user add new songs to the library without restarting the client.

**Preconditions:**

1. The jMPD is listening on the configured port.
2. The database is accessible.
3. The jMPC is connected to jMPD on the same configured port.

**Scenario:**

| **Action** | **Software Reaction** |
| --- | --- |
| 1. User clicks “Options” menu | 1. Client populates and expands the options menu. |
| 2. User points cursor to the "update" item | 2. Client highlights the item with a contrasting background color. |
| 3. User clicks "Update Database" item | 3a. Server updates database  3b. Client updates view with new information |

**Scenario Notes:**

If the server returns and empty list in step 3, the client does nothing.

**Post Conditions:**

1. The daemon's database has been changed to reflect the changes to the music directory.
2. The client's GUI reflects the changes in the database.

**Exceptions:**

1. The database cannot be accessed
2. The music directory cannot be accessed
3. The daemon is not running
4. The client is not connected to the same port as jMPD.

**Required GUI:**

1. jMPC\_Main\_View
2. jMPC\_Menu
3. jMPC\_Library\_View

**Use Cases Utilized:**

None

**Timing Constraints:**

None

**Use Case 6: UserAddsTrackToPlayQueue**

**Overview:**

Allows user to append new tracks to the end of the play queue.

**Preconditions:**

1. The jMPD is listening on the configured port.
2. The database is accessible.
3. The jMPC is connected to jMPD on the same configured port.
4. User is looking at jMPC\_Library\_View.

**Scenario:**

| **Action** | **Software Reaction** |
| --- | --- |
| 1. User double clicks on a track in the library view. | 1a. Client sends the "ADD" command and the track id to the daemon.  1b. Server adds specified track to play queue, returns the updated play queue information.  1c. Client updates the playqueue tab to reflect the changes to the queue. |

**Scenario Notes:**

None.

**Post Conditions:**

1. The daemon's play queue now contains the user selected tracks.
2. The client's GUI reflects the changes in the play queue.

**Exceptions:**

1. The database cannot be accessed
2. The music directory cannot be accessed
3. The daemon is not running
4. The client is not connected to the same port as jMPD.

**Required GUI:**

1. jMPC\_Main\_View
2. jMPC\_Library\_View

**Use Cases Utilized:**

None

**Timing Constraints:**

None

**Use Case 7: UserRemovesTrackFromPlayQueue**

**Overview:**

Allows user to remove tracks from the play queue.

.

**Preconditions:**

1. The jMPD is listening on the configured port.

2. The database is accessible.

3. The jMPC is connected to jMPD on the same configured port.

4. User is looking at jMPC\_Play\_Queue\_View.

**Scenario:**

| **Action** | **Software Reaction** |
| --- | --- |
| 1. User double clicks on a track in the play queue tab | 1a.Client sends the “REM” command and to the track id to the daemon.  1b. Server removes the specified track from the play queue and returns the updated play queue information.  1c. Client updates the play queue tab to reflect the changes to the queue. |

**Scenario Notes:**

None.

**Post Conditions:**

1. The daemon's play queue no longer contains the user selected tracks.

2. The client's GUI reflects the changes in the play queue.

**Exceptions**:

1. The database cannot be accessed

2. The music directory cannot be accessed

3. The daemon is not running

4. The client is not connected to the same port as jMPD.

**Required GUI:**

1. MainViewController

**Use Cases Utilized:**

None

**Timing Constraints:**

None

**Use Case 8: UserClearsPlayQueue**

**Overview:**

Allows user to remove all tracks from the play queue.

.

**Preconditions:**

**1. The jMPD is listening on the configured port.**

**2. The database is accessible.**

**3. The jMPC is connected to jMPD on the same configured port.**

**Scenario:**

| **Action** | **Software Reaction** |
| --- | --- |
| 1. User clicks on clear play queue button in the play queue tab | 1a.Client sends the “CLEAR” command to the daemon.  1b. Server removes all the tracks in the play queue and returns the updated play queue information.  1c. Client updates the play queue tab to reflect the changes to the queue. |

**Scenario Notes:**

If the play queue is empty, the client does not send the clear command.

**Post Conditions:**

1. The daemon's play queue no longer contains any tracks.

2. The client's GUI reflects the changes in the play queue.

**Exceptions**:

1. The daemon is not running

2. The client is not connected to the same port as jMPD.

**Required GUI:**

1. jMPC\_Main\_View

**Use Cases Utilized:**

None

**Timing Constraints:**

None

**Use Case 9: UserBrowsesLibrary**

**Overview:**

Allows user to browse the library.

.

**Preconditions:**

**1. The jMPD is listening on the configured port.**

**2. The database is accessible.**

**3. The jMPC is connected to jMPD on the same configured port.**

**Scenario:**

|  |  |
| --- | --- |
| **Action** | **Software Reaction** |
| 1. **User clicks on an artist name in library view** | **1.** **Client highlights the artist entry with a contrasting background color, and loads that artist's albums in the album pane. And Shows all songs by that artist in the tracks pane.** |
| 1. **User clicks on an album name in the library pane** | **2. Client highlights the album entry with a contrasting background color, and loads just that album's list of tracks in the song pane.** |

**Scenario Notes:**

None

**Post Conditions:**

1. The daemon's play queue no longer contains any tracks.

2. The client's GUI reflects the changes in the play queue.

**Exceptions**:

1. The client GUI displays the information requested by the user sorted alphabetically and by track number.

**Required GUI:**

1. MainViewController

**Use Cases Utilized:**

None

**Timing Constraints:**

None

**Use Case 10: UserCreatesNewPlaylist**

**Overview:**

Allows user to create a playlist.

.

**Preconditions:**

**1. The jMPD is listening on the configured port.**

**2. The database is accessible.**

**3. The jMPC is connected to jMPD on the same configured port.**

**4. The playlist directory is writable.**

**Scenario:**

|  |  |
| --- | --- |
| **Action** | **Software Reaction** |
| **1. User clicks on the Add New Playlist button in the left pane.** | **1. Client opens a text entry view** |
| **2. User types in a string of characters for the name of the playlist.** | **2. Client echos their input in the text entry box.** |
| **3. User presses enter, or clicks the "Done" button** | **3a. Client sends the "NEW" command to the daemon along with the user string.**  **3b.** **Server receives command message**  **3c. Server parses command message, extracts "NEW" command and the playlist name string.**  **3d. Server creates an empty playlist in the predefined playlist directory with the given name, returns "DONE" message to the client.**  **3e. Client displays the new entry in the left pane, under the Playlist header** |

**Scenario Notes:**

None

**Post Conditions:**

**1. The client GUI displays the information requested by the user.**

**Exceptions**:

1. The daemon is not running.

2. The client is not connected to the same port as jMPD.

3. The user string matches a playlist that already exists. Displays error message, allows the user to retry.

**Required GUI:**

1. jMPC\_Text\_Entry\_View

2. jMPC\_Library\_View or jMPC\_Play\_Queue\_View

**Use Cases Utilized:**

None

**Timing Constraints:**

None

**Use Case 11: UserDeletesPlaylist**

**Overview:**

Allows user to delete a playlist.

**Preconditions:**

**1. The jMPD is listening on the configured port.**

**2. The database is accessible.**

**3. The jMPC is connected to jMPD on the same configured port.**

**4. The playlist directory is writable.**

**Scenario:**

|  |  |
| --- | --- |
| **Action** | **Software Reaction** |
| **1. User right clicks on a playlist entry in the left pane.** | **1. Client opens a context menu for playlist** |
| **2. User clicks "Delete" context menu item.** | **2a. Client sends the "DEL" command to the daemon, along with the playlist name**  **2b.** **Server receives command message**  **2c. Server parses command message, extracts "DEL" command and the playlist name string**  **2d.** **Server finds the playlist with the given name, deletes it, and returns "DONE" message to the client**  **2e.** **Client removes the corresponding playlist entry from the left pane** |

**Scenario Notes:**

None

**Post Conditions:**

**1. The client GUI displays the information requested by the user.**

**Exceptions**:

1. The daemon is not running.

2. The client is not connected to the same port as jMPD.

3. The playlist directory is not writable.

**Required GUI:**

1. jMPC\_Main\_View

2. jMPC\_Playlist\_Context\_Menu

**Use Cases Utilized:**

None

**Timing Constraints:**

None

**Use Case 12: UserAddsToPlaylist**

**Overview:**

Allows user to add tracks to a playlist.

**Preconditions:**

**1. The jMPD is listening on the configured port.**

**2. The database is accessible.**

**3. The jMPC is connected to jMPD on the same configured port.**

**4. The playlist directory is writable.**

**Scenario:**

|  |  |
| --- | --- |
| **Action** | **Software Reaction** |
| **1.User right clicks on a track entry in the library view** | **1. Client highlights the track entry with a contrasting background color and display a drop down menu** |
| 1. **User selects add to play list option** | **2. An entry field dialog box appears with a list of the playlist.** |
| 1. **User selects the playlist of choice** | **3a.Client sends “ADDTOPLAYLIST” to daemon with track id**  **3b. Server adds track to playlist and returns the update playlist information.**  **3c. Client display updated playlist** |

**Scenario Notes:**

The user is allowed to multi-select tracks in the library. They will be placed into the playlist in ascending id order

**Post Conditions:**

**1. The client GUI displays the information requested by the user.**

**Exceptions**:

1. The daemon is not running.

2. The client is not connected to the same port as jMPD.

3. The playlist directory is not writable.

**Required GUI:**

1. jMPC\_Library\_View

2. jMPC\_Playlist\_Select\_View

**Use Cases Utilized:**

None

**Timing Constraints:**

None

**Use Case 13: UserRemovesFromPlaylist**

**Overview:**

Allows user to remove tracks from a playlist.

**Preconditions:**

**1. The jMPD is listening on the configured port.**

**2. The database is accessible.**

**3. The jMPC is connected to jMPD on the same configured port.**

**4. The playlist directory is writable.**

**Scenario:**

|  |  |
| --- | --- |
| **Action** | **Software Reaction** |
| **1.User right clicks on a track entry in the library view** | **1. Client highlights the track entry with a contrasting background color and display a drop down menu** |
| **2.User selects remove from play list option** | **2a.Client sends “REMFROMPLAYLIST” to daemon with track id**  **2b. Server removes track from playlist and returns the update playlist information.**  **2c. Client displays updated playlist** |

**Scenario Notes:**

The user is allowed to multi-select tracks in the playlist. They will all be removed from the playlist

**Post Conditions:**

**1. The client GUI displays the updated playlist information.**

**2. The playlist file no longer contains the deleted entries**

**Exceptions**:

1. The daemon is not running.

2. The client is not connected to the same port as jMPD.

3. The playlist directory is not writable.

**Required GUI:**

1. jMPC\_Library\_View

2. jMPC\_Playlist\_Select\_View

**Use Cases Utilized:**

None

**Timing Constraints:**

None

**Use Case 14: UserSavesPlayQueueToPlaylist**

**Overview:**

Allows user to save the current play queue as a playlist.

**Preconditions:**

**1. The jMPD is listening on the configured port.**

**2. The database is accessible.**

**3. The jMPC is connected to jMPD on the same configured port.**

**4. The playlist directory is writable.**

**5. The play queue is not empty**

**Scenario:**

|  |  |
| --- | --- |
| **Action** | **Software Reaction** |
| **1.User right clicks on a track entry in the play queue** | **1. Client highlights the track entry with a contrasting background color and display a drop down menu** |
| **2.User selects save play queue** | **2a.Client sends “SAVEPLAYQUEUE” to daemon with the new playlist name**  **2b. Server creates a new playlist and returns the update playlist**  **2c. Client displays updated playlist** |

**Scenario Notes:**

The user is allowed to multi-select tracks in the playlist. They will all be removed from the playlist

**Post Conditions:**

**1. The client GUI displays the updated playlist information.**

**2. The playlist file no longer contains the deleted entries**

**Exceptions**:

1. The daemon is not running.

2. The client is not connected to the same port as jMPD.

3. The playlist directory is not writable..

**Required GUI:**

1. jMPC\_Main\_View

2. jMPC\_Playlist\_Select\_View

**Use Cases Utilized:**

None

**Timing Constraints:**

None

**Use Case 15: UserConfiguresDaemonSettings**

**Overview:**

Allows user to configure the jMPD settings.

**Preconditions:**

**1. The jMPD is listening on the configured port.**

**2. The config file is writable.**

**3. The jMPC is connected to jMPD on the same configured port**

**Scenario:**

|  |  |
| --- | --- |
| **Action** | **Software Reaction** |
| **1.** **User clicks "daemon" menu bar entry** | **1. Client populates and expands the daemon menu** |
| **2.** **User points cursor to the "settings" item** | **2.Client highlights the item with a contrasting background color** |
| **3. User clicks "settings" item** | **3. Client opens and populates the jMPC\_Daemon\_Config\_View** |
| **4. User clicks one of the textboxes in the config view** | **4. A text cursor begins blinking in the clicked textbox.** |
| **5. User edits the text in the textbox** | **5. The client echos the user's edits.** |
| **6. User clicks the "Save" button in the jMPC\_Daemon\_Config\_View** | **6. Client writes the changes to the config file.** |

**Scenario Notes:**

If the "Cancel" button is instead pressed, the config file is left unchanged

**Post Conditions:**

**1. The daemon's configuration file now contains the updated information**

**Exceptions**:

1.The config file is not writable

**Required GUI:**

1. jMPC\_Main\_View

2. jMPC\_Menu\_Bar

3. jMPC\_Menu\_Daemon\_Entry

4. jMPC\_Daemon\_Config\_View

**Use Cases Utilized:**

None

**Timing Constraints:**

None

**Use Case 16: UserConfiguresClientSettings**

**Overview:**

Allows user to configure the jMPC settings.

**Preconditions:**

**1. The client config file is writable**

**Scenario:**

|  |  |
| --- | --- |
| **Action** | **Software Reaction** |
| **1.User clicks "client" menu bar entry** | **1. Client populates and expands the client menu.** |
| **2.** **User points cursor to the "settings" item** | **2.Client highlights the item with a contrasting background color** |
| **3. User clicks "settings" item** | **3. Client opens and populates the jMPC\_Config\_View** |
| **4. User clicks one of the textboxes in the config view** | **4. A text cursor begins blinking in the clicked textbox..** |
| **5. User edits the text in the textbox** | **5. The client echos the user's edits.** |
| **6. User clicks the "Save" button in the jMPC\_Daemon\_Config\_View** | **6. Client writes the changes to the client's config file** |

**Scenario Notes:**

The configuration view may also contain checkboxes, which control togglable settings such as "Cache Retrieved album art". If the "Cancel" button is instead pressed, the config file is left unchanged.

**Post Conditions:**

**1. The client's configuration file now contains the updated information Exceptions**:

1. The config file is not writable

**Required GUI:**

1. jMPC\_Main\_View

2. jMPC\_Menu\_Bar

3. jMPC\_Menu\_Client\_Entry

4. jMPC\_Config\_View

**Use Cases Utilized:**

None

**Timing Constraints:**

None