| **Use Case ID** | UC 6 |
| --- | --- |
| **Use Case Name** | User adds a song to the queue |
| **Author Name** | Shane Arwood |
| **Primary Actor** | Subscriber |
| **Actor’s Goal** | User wants to play a specific song after the current song finishes |
| **Business Value** | IMPORTANT |

## Description

The use case details the event that a subscriber is listening to their current queue of songs and wants to add a new song to the queue. The system checks if the song is already in the queue and also allows the user to select which part of the queue they want to add the song to.

## Main Success Scenario

| **Step** | **Actor Action** | **System Response** |
| --- | --- | --- |
| **1** | User selects a song and clicks “Add to queue” | System confirms that the song is not already in the queue. |
| **2** |  | System prompts the user to choose whether to add the song to the beginning of the queue, end of the queue, or after a specified song. |
| **3** | User chooses to add the song to the beginning of the queue, end of the queue, or selects the song after which the new song should play. | System closes the dialogue box. |
| **4** |  | System adds the song to the specified position. |
| **5** |  | System plays the queue in order with the new song added. |

## Alternative Success Scenario 1: Song is already in queue and the user decides to add it anyways.

**Trigger Condition**: When the user selects a song to add to the queue, that song is already found in the queue.

| **Step** | **Actor Action** | **System Response** |
| --- | --- | --- |
| **1** | User selects a song and clicks “Add to queue” | System checks that the song is already in the queue and displays “Song already in queue, add anyways?” |
| **2** | User selects “Yes” | System prompts the user to choose whether to add the song to the beginning of the queue, end of the queue, or after a specified song. |
| **3** | User chooses to add the song to the beginning of the queue, end of the queue, or selects the song after which the new song should play. | System closes the dialogue box. |
| **4** |  | System adds the song to the specified position. |
| **5** |  | System plays the queue in order with the new song added. |

## Alternative Failure Scenario 2: Song is already in queue and the user declines to add it again.

**Trigger Condition**: When the user selects a song to add to the queue, that song is already found in the queue.

| **Step** | **Actor Action** | **System Response** |
| --- | --- | --- |
| **1** | User selects a song and clicks “Add to queue” | System checks that the song is already in the queue and displays “Song already in queue, add anyways?” |
| **2** | User selects “No” | System closes the dialogue box. |
| **3** |  | System does not add the song to the queue. |
| **4** |  | System plays the queue in order without the song added. |

| **Secondary Actors** | N/A |
| --- | --- |
| **Trigger Event** | User selected a song and clicked “Add to queue” |
| **Precondition** | A song is already playing, and there are already other songs in the queue |
| **Frequency of Execution** | Regularly (several times a day) |
| **Success Criteria (optional)** | The queue will play in the correct order with the new song added in the user’s specified position |
| **Development Cost** | AVERAGE |

## Discussion

N/A

## Issues / Risks

N/A

## Non-Functional Requirements

The processing of the user choices (double checking if the song is already in the queue and choosing position to add song in the queue) should be almost instantaneous so there is no delay in the system playing the songs in the correct order.

## UML Activity Diagram

