Use Cases

for

Collaborative Spotify Queue

**Version 2.10**

**Prepared by Team 7**

**March 10th, 2016**

Use Case List

|  |  |  |
| --- | --- | --- |
| ***ID*** | ***Primary Actor*** | ***Use Case Title*** |
| 1.0 | Spotify Queue User | Log into Spotify via OAuth |
| 2.0 | Spotify Queue Host | Create Party Group |
| 3.0 | Spotify Queue User | Search for Party Group |
| 4.0 | Spotify Queue User | Search for Specific Spotify Song |
| 5.0 | Spotify Queue User | Add Specific Spotify Song to Queue |
| 6.0 | Spotify Queue User | Vote for Song in Queue |
| 7.0 | Spotify Queue User | End Party and Send Group’s Youtube Playlist |

Use Case 1.0

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1.0 | | |
| Use Case Name: | Log into Spotify via OAuth | | |
| Created By: | Dan Brown | Last Updated By: |  |
| Date Created: | 03/11/2016 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actors: | Spotify Queue User |
| Description: | The Spotify Queue User is whoever is using our web application for the purpose of viewing, adding, or voting for songs in a party group’s Spotify queue.The purpose of this use case is so that a Spotify Queue User can log in and gain access to the features of our application; this also prevents the manipulation of restricted data and resources to be carried out by unauthorized and/ or malicious clients. |
| Trigger: | Spotify Queue User wants to log into Spotify [External] |
| Preconditions: | 1. The Spotify Queue User’s username and password combination already exists (and is valid) 2. System can connect to Spotify API 3. User’s computer has sufficient free memory available to launch task |
| Postconditions: | 1. The user is greeted with a welcome message and gains access to other states of the application (or if not authenticated, then a message to try logging in again is displayed) |
| Normal Flow: | 1.0 Type in username and password  1.1 Hit Spotify API via OAuth to login (be authenticated)  1.2 User is logged in (and now has access to the application’s features) |
| Alternative Flows: | 1.2 user is not logged in (due to wrong password or because username is not found) |
| Exceptions: | 1.0.E.1 Spotify API login request via OAuth returns an unexpected error, then ask potential user to try logging in again  1.0.E.2 Spotify API login request via OAuth times out (> 30 seconds), then ask potential user to try logging in again |
| Includes: | N/A |
| Priority: | High priority, considered a core functionality |
| Frequency of Use: | At least once per day, since that will be the maximum session time before a user is logged out and must login again. |
| Business Rules: | 1. A Spotify Queue User must type in valid information into the user and password fields (any fields that contain special/ unique characters or are left blank will not be processed correctly, leading to an automatic login fail) 2. A Spotify Queue User must wait for any previous login attempt(s) to finish before trying to login again (this is to avoid excess server load and to prevent a user from getting logged in multiple times, which could lead to unforeseen bugs) |
| Special Requirements: | 1. The system must respond to the login request within a few minutes (user must be either be authenticated and logged in or refused and asked to try again, within 180 seconds of making the request). |
| Assumptions: | Spotify API is accessible |
| Notes and Issues: | N/A |

Use Case 2.0

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2.0 | | |
| Use Case Name: | Create Party Group | | |
| Created By: | Dan Brown | Last Updated By: |  |
| Date Created: | 03/11/2016 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actors: | Spotify Queue User |
| Description: | The Spotify Queue User is whoever is using our web application for the purpose of viewing, adding, or voting for songs in a party group’s Spotify queue. The purpose of this use case is so that a user can create a “party group” (a set of members that will have access to their specific Spotify Queue as well as the songs associated with that queue, and will have the ability to vote, add, and later, to export the songs used in that queue as a youtube playlist). |
| Trigger: | User wants to create a “party group” [External] |
| Preconditions: | 1. User’s identity has been authenticated 2. User’s computer has sufficient free memory available to launch task 3. User has the correct privileges in order to initiate this task |
| Postconditions: | 1. A group page/ state is displayed, showing the current users that are part of the group, as well as the party group ID 2. Other users can now join the party group |
| Normal Flow: | 2.0 Tap/ click button to create new party group  2.1 New party group (along with a new party group ID) is created |
| Alternative Flows: | 2.1 new party group is not created, because user is already in a party group |
| Exceptions: | 2.0.E.1 System is unable to create a new party group; ask user to try again |
| Includes: | Use case 1.0 for Spotify user authentication |
| Priority: | High priority, considered a core functionality |
| Frequency of Use: | Whenever a group wants a Spotify party queue playlist. We expect it to be a few times for a normal weekday, with more groups being created over the weekend nights. |
| Business Rules: | 1. A user must wait for the party group to be created (or fail at being created), before attempting to try again (this is to lessen the load on our system) |
| Special Requirements: | 1. The creation of a party group should not take a long time (less than 30 seconds) 2. Must use and communicate with some form of a database (using SQL or NoSQL) |
| Assumptions: | The user has the ability to click/ tap to initiate the command to create a new party group |
| Notes and Issues: | N/A |

Use Case 3.0

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3.0 | | |
| Use Case Name: | Search For Party Group | | |
| Created By: | Wendy Qian | Last Updated By: | Dan Brown |
| Date Created: | 03/10/2016 | Date Last Updated: | 03/11/2016 |

|  |  |
| --- | --- |
| Actors: | Spotify Queue User |
| Description: | The Spotify Queue User is whoever is using our web application for the purpose of viewing, adding, or voting for songs in a party group’s Spotify queue. This purpose of this use case is so that the user can search for a specific party group, so that they can add songs to that party’s queue, vote on the party’s songs, and receive an email with the YouTube playlist of the songs after the party ends. |
| Trigger: | User wants to join a party group [External] |
| Preconditions: | 1. User’s identity has been authenticated 2. User has access to a verified party group’s ID number 3. User’s computer has sufficient free memory available to launch task 4. System can connect to Spotify API |
| Postconditions: | 1. The party group search provides the user with the desired party, or with a “none found” statement if that party ID does not exist |
| Normal Flow: | 3.0 Type party group ID number into party group search bar  3.1 Access the party group database to locate the desired party  3.2 Return results of search  3.3 User joins party |
| Alternative Flows: | 3.2 Party group ID Number is not found in database |
| Exceptions: | 3.0.E.1 An unexpected error occurs, then asks the user to try entering in the ID number again  3.0.E.2 Party group ID number retrieval from the database times out (> 30 seconds), then asks user to try again |
| Includes: | Use case 1.0 for Spotify user authentication |
| Priority: | Highest priority, considered a core functionality |
| Frequency of Use: | Once per party group |
| Business Rules: | 1. A user must terminate or wait until their previous search is finished before starting a new search (so that there is less load on the system) 2. A user’s search terms must not have invalid characters (characters that are not letters or numbers will not return any valid party groups) |
| Special Requirements: | 1. The results of the search should not take a long time (less than 30 seconds to find the party with the respective ID number) 2. Executing a search again for the same part group ID number, should return identical results 3. Must use and communicate with some form of a database (using SQL or NoSQL) |
| Assumptions: | 1. The Spotify Queue User will input the correct party group ID number to the best of their ability 2. The Spotify Queue User wants to join the party that they are searching for |
| Notes and Issues: | N/A |

Use Case 4.0

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 4.0 | | |
| Use Case Name: | Search For Specific Spotify Song | | |
| Created By: | Wendy Qian | Last Updated By: | Dan Brown |
| Date Created: | 02/12/2016 | Date Last Updated: | 2/13/16 |

|  |  |
| --- | --- |
| Actors: | Spotify Queue User |
| Description: | The Spotify Queue User is whoever is using our web application for the purpose of viewing, adding, or voting for songs in a party group’s Spotify queue. This purpose of this use case is so that the user can search for a specific Spotify song, for the later purpose of adding it to the queue. |
| Trigger: | User wants to search for a song to add to the Spotify queue [External] |
| Preconditions: | 1. User’s identity has been authenticated 2. User is associated with a verified party group 3. User’s computer has sufficient free memory available to launch task 4. System can connect to Spotify API |
| Postconditions: | 1. The search provides the user with either options of songs corresponding to the search input, or with a “none found” statement |
| Normal Flow: | 4.0 Type search into search bar  4.1 Hit Spotify API to perform search  4.2 Return results of search |
| Alternative Flows: | 4.2 Song is not found in Spotify database |
| Exceptions: | 4.0.E.1 Spotify API request returns an unexpected error, then ask user to try again  4.0.E.2 Spotify API times out (> 30 seconds), then ask user to try again |
| Includes: | Use case 1.0 for Spotify user authentication |
| Priority: | Highest priority, considered a core functionality |
| Frequency of Use: | At least 24 times per hour, considering songs are on average about 2.5 minutes long |
| Business Rules: | 1. A user must terminate or wait until their previous search is finished before starting a new search (so that there is less load on the system) 2. A user’s search terms must be part of actual song titles (gibberish or any incoherent search terms will lead to bad results) |
| Special Requirements: | 1. The results of the search should not take a long time (less than a few minutes for each queried song) 2. Executing a search again for the same song, should return similar (if not identical) results 3. Must use and communicate with some form of a database (using SQL or NoSQL) |
| Assumptions: | Spotify API is accessible and can be used to search for a specific song |
| Notes and Issues: | N/A |

Use Case 5.0

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 5.0 | | |
| Use Case Name: | Add Specific Spotify Song to Queue | | |
| Created By: | Wendy Qian | Last Updated By: |  |
| Date Created: | 03/10/2016 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actors: | Spotify Queue User |
| Description: | The Spotify Queue User is whoever is using our web application for the purpose of viewing, adding, or voting for songs in a party group’s Spotify queue. This purpose of this use case is so that the user can add a specific Spotify song to the party queue, after searching for it (as described in Use Case 4.0). This is a separate use case from Use Case 4.0 because users in Use Case 4.0 can search for songs, but not necessarily add them to the party’s song queue. |
| Trigger: | User wants to add a song to the Spotify queue [External] |
| Preconditions: | 1. User’s identity has been authenticated 2. User is associated with a verified party group 3. User has searched for Spotify songs in the search bar 4. User’s computer has sufficient free memory available to launch task 5. System can connect to Spotify API |
| Postconditions: | 1. The song that the user chose to add shows up in the party group’s song queue |
| Normal Flow: | 5.0 Select “Add Song” in the search results from use case 4.0  5.1 Hit the Spotify API to access the song data  5.2 Add the song to the party group’s song queue by transferring the retrieved Spotify song data to the queue playlist |
| Alternative Flows: | 5.2 If the song is already in the party group’s song queue, an error message is shown to the user, and the song is not added to the queue |
| Exceptions: | 5.0.E.1 Spotify API request returns an unexpected error, then ask user to try again  5.0.E.2 Spotify API times out (> 30 seconds) when adding the song, then ask user to try again |
| Includes: | Use case 1.0 for Spotify user authentication  Use case 4.0 for searching for specific Spotify song |
| Priority: | Highest priority, considered a core functionality |
| Frequency of Use: | At least 24 times per hour, considering songs are on average about 2.5 minutes long |
| Business Rules: | 1. A user must wait for their previous request to add a song is finished before adding a new song (so that there is less load on the system) |
| Special Requirements: | 1. Adding a chosen song to the party group’s song queue should not take a long time (less than 30 seconds for each chosen song) 2. Must use and communicate with some form of a database (using SQL or NoSQL) |
| Assumptions: | Spotify API is accessible and can be used to search for a specific song |
| Notes and Issues: | N/A |

Use Case 6.0

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 6.0 | | |
| Use Case Name: | Vote for Song in Queue | | |
| Created By: | Molly Shopper | Last Updated By: |  |
| Date Created: | 03/12/2016 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actors: | Spotify Queue User |
| Description: | User wants to upvote or downvote a specific song in the queue, which will in turn move that song up or down in the playlist queue. |
| Trigger: | User wants to vote for a song [External] |
| Preconditions: | 1. User’s identity has been authenticated 2. User is associated with a verified party group 3. There is at least one song in the playlist 4. User’s computer has sufficient free memory available to launch task 5. System can connect to Spotify API |
| Postconditions: | 1. The number of upvotes or downvotes increases or decreases accordingly 2. Song moves within queue accordingly |
| Normal Flow: | 6.0 Tap/click button to upvote or downvote  6.1 Song moves to new spot in queue if appropriate |
| Alternative Flows: | None |
| Exceptions: | 6.0.E.1 System is unable to process voting; ask user to try again |
| Includes: | Use case 1.0 for Spotify user authentication |
| Priority: | Medium priority, not vital for app to work. |
| Frequency of Use: | Very frequent but only when a party group is active. Likely a few times per minute. |
| Business Rules: | 1. User must wait to vote again until the first voting has finished |
| Special Requirements: | 1. Voting should not take a long time (less than a second) |
| Assumptions: | The user has the ability to click/tap to initiate the command |
| Notes and Issues: | N/A |

Use Case 7.0

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 7.0 | | |
| Use Case Name: | End Party and Send Group’s Youtube Playlist | | |
| Created By: | Molly Shopper | Last Updated By: |  |
| Date Created: | 03/12/2016 | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actors: | Spotify Queue User |
| Description: | Once the party is over, no more songs may be added or voted on. A Youtube playlist of the songs from the collaborative Spotify playlist is created and sent to the users via the app. |
| Trigger: | Party ends [External] |
| Preconditions: | 1. User’s identity has been authenticated 2. User is associated with a verified party group 3. There is at least one song in the playlist 4. User’s computer has sufficient free memory available to launch task 5. System can connect to Spotify and Youtube APIs |
| Postconditions: | 1. All members of party group receive Youtube playlist link |
| Normal Flow: | 7.0 Host clicks button to end party  7.1 Hit Youtube API to find music video for each song in playlist  7.2 Create Youtube playlist with these music videos  7.3 Send link to everyone in party group |
| Alternative Flows: | 7.1 There are no songs in playlist, error message is displayed and no Youtube playlist is created |
| Exceptions: | 7.1.E.1 Song is not in Youtube API, error message is displayed but system continues to make playlist with other songs |
| Includes: | Use case 1.0 for Spotify user authentication |
| Priority: | High priority, one of app’s main functions. |
| Frequency of Use: | Once per party |
| Business Rules: | 1. Once the party is over, a group cannot reopen the party, must create a new one |
| Special Requirements: | 1. Creating the playlist should not take longer than 30 seconds |
| Assumptions: | The user has the ability to click/tap to initiate the command and Youtube API is accessible |
| Notes and Issues: | N/A |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Molly Shopper | 2/13/16 | Updated use cases and added normal flow, alternative flow, and frequency of use to table | 1.10 |
| Mike Gajda | 2/13/16 | Updated use cases, defined Expectations, Includes and Priority | 1.11 |
| Dan Brown | 2/13/16 | Updated business rules, special requirements, assumptions, and preconditions | 1.12 |
| Wendy Qian | 2/15/16 | Made formatting edits, updated includes | 1.13 |
| Wendy Qian | 3/10/16 | Created use cases 3.0 and 5.0, edited “Includes” in use case 4.0 | 2.10 |
| Dan Brown | 3/11/16 | Created use cases 1.0 and 2.0, updated assumptions for use case 3.0 | 2.11 |
| Molly Shopper | 3/12/16 | Created use cases 6.0 and 7.0 | 2.12 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |