User Story Descriptions

The following tables accompany the use case diagrams provided elsewhere in the documentation. They serve to provide further detail regarding the use cases illustrated. See also the SRS document.

Use Case: Search For Recommendations

|  |  |
| --- | --- |
| **Use Case Name** | Search for Recommendations |
| **XRef** | Section 2.1 |
| **Trigger** | The active user, after logging in, requests a recommendation. |
| **Precondition** | The active user has music history data stored within the system, which can be referenced to produce recommendations. |
| **Basic Path** | 1. The system identifies the user and locates listening history. 2. Using collaborative filtering, the system compares active user music data with other users’ music data to produce song recommendations. 3. The system provides the active user with an output list of song titles and corresponding artists. |
| **Alternative Paths** | \*\*Potententially: If the user chooses, he or she may enter a specific song and receive recommendations. |
| **Postcondition** | The user receives recommendations. |
| **Exception Paths** | The attempt may be abandoned at any time. |
| **Other** | None |

Use Case: Login

|  |  |
| --- | --- |
| **Use Case Name** | Login |
| **XRef** | Section 2.1 |
| **Trigger** | The active user enters the system by entering a username and password. |
| **Precondition** | The active user has signed up as a user, and his or her information is in the database. |
| **Basic Path** | 1. The active user enters his or her username into the system. 2. The active user enters his or her password into the system. 3. The system identifies the user, using the information in the database. 4. If the user has not yet signed up and therefore is not in the database, the system will display a message. 5. If the user is in the database, the system will display a page where the user may take further action. |
| **Alternative Paths** | N/A |
| **Postcondition** | The user is able to view and interact with the system. |
| **Exception Paths** | The user may not have signed up, so his information may not be in the system. |
| **Other** | If user has not signed up, prompt that user to signup page. |

Use Case: Logout

|  |  |
| --- | --- |
| **Use Case Name** | Logout |
| **XRef** | Section 2.1 |
| **Trigger** | The active user leaves the system by clicking logout button. |
| **Precondition** | The active user is logged in currently. |
| **Basic Path** | 1. The active user clicks the logout button. 2. The system ends the user’s session and displays the home view (login page). |
| **Alternative Paths** | The session may expire after a period of time, logging the user out (without button trigger) |
| **Postcondition** | The login view is shown. The active user is no longer logged in. |
| **Exception Paths** | N/A |
| **Other** | None |

Use Case: Subscribe to Artist

|  |  |
| --- | --- |
| **Use Case Name** | Subscribe to Artist |
| **XRef** | Section 2.1 |
| **Trigger** | The active user searches an artist and chooses to follow (subscribe) to that artist. |
| **Precondition** | The active user has not already subscribed to the particular artist. The active user is logged in. |
| **Basic Path** | 1. The active user searches an artist. 2. The active user selects an artist and selects the “follow” button. |
| **Alternative Paths** |  |
| **Postcondition** | The user receives random song recommendations from that artist whenever he or she logs in. |
| **Exception Paths** | The attempt may be abandoned at any time. |
| **Other** | None |

Use Case: Signup

|  |  |
| --- | --- |
| **Use Case Name** | Signup |
| **XRef** | Section 2.1 |
| **Trigger** | The active user registers with the system. |
| **Precondition** | The active user does not already have an account with the system. |
| **Basic Path** | 1. The active user selects “signup” 2. The active user enters his or her information into the system. 3. The system stores the active user’s information in the database for future login. |
| **Alternative Paths** | N/A |
| **Postcondition** | The user now has an account and may use the system. |
| **Exception Paths** | The attempt may be abandoned at any time. |
| **Other** | None |

Use Case: Add Song to Database

|  |  |
| --- | --- |
| **Use Case Name** | Add Song to Database |
| **XRef** | Section 2.1 |
| **Trigger** | The administrator logs in and selects “add song.” |
| **Precondition** | The administrator must be logged in. |
| **Basic Path** | 1. The administrator logs in using his or her username and password. 2. The administrator selects “add song.” 3. The administrator enters song information. 4. The system saves the song information in the database. 5. The system returns to the home page. |
| **Alternative Paths** | N/A |
| **Postcondition** | The song is now in the system database. |
| **Exception Paths** | The attempt may be abandoned at any time. |
| **Other** | None |

Use Case: Delete Song from Database

|  |  |
| --- | --- |
| **Use Case Name** | Delete Song from DataBase |
| **XRef** | Section 2.1 |
| **Trigger** | The administrator logs in and selects “delete song.” |
| **Precondition** | The administrator must be logged in. The song must be in the database. |
| **Basic Path** | 1. The administrator logs in using his or her username and password. 2. The administrator selects “delete song.” 3. The administrator enters song title. 4. The system checks for the song in the database. 5. If the song is not in the database, the system displays a message. 6. If the song is in the database, the system prompts the administrator with a warning message that the action cannot be undone. 7. If the administrator accepts the warning message, the system deletes the song from the database. 8. The system returns to the homepage. |
| **Alternative Paths** | N/A |
| **Postcondition** | The song is no longer in the databse. |
| **Exception Paths** | The attempt may be abandoned at any time. |
| **Other** | None |