**Jeremy Hunat**

**Use Cases**

**ID: UC\_01**

**Title:** Log in to Spotify

**Description:** Users will be prompted to log-in to their Spotify account and give the app permission to connect to their account.

**Primary Actor:** User

**Precondition:** None

**Minimal Guarantees:** User would need to sign-up for an account, and give

**Success Guarantees:** User successful give the app access to permissions.

**Trigger:** Either clicking on host or join a session.

**Main Success Scenario:** The app would be able to communicate with the users account through the Spotify API.

**Extensions:**

* User closes the authentication without logging in.
* User does not give permission to the app.
* User provides an invalid log in.

**ID: UC\_02**

**Title:** Host a listening session.

**Description:** User will be able to create a private listening room and can invite people to join using a private randomly generated key.

**Primary Actor:** User

**Precondition:** User would need to be logged in to their Spotify account.

**Minimal Guarantees:** A room is created.

**Success Guarantees:** A room is created, and people are able to join off the given code.

**Trigger:** Clicking on the “Host a session” button.

**Main Success Scenario:** A private room is created, and the most recent song played in Spotify will be played automatically by default.

**Extensions:**

* Connection is lost when trying to create a room.
* Authentication for the Spotify account has expired.

**ID: UC\_03**

**Title:** Join a listening session.

**Description:** User

**Primary Actor:** User

**Precondition:** User would need to be logged in to their Spotify account.

**Minimal Guarantees:** User is able to join a private room.

**Success Guarantees:** User is able to join a private room, and music playing will be synced up to the current progress of the host.

**Trigger:** Clicking on the “Join session” button.

**Main Success Scenario:** User is able to join a private room, and music playing will be synced up to the current progress of the host.

**Extensions:**

* User inputs an invalid private key code.
* User enters a private room at full capacity.
* User enters a room that no longer exists.

**ID: UC\_04**

**Title:** Operate media controls.

**Description:** The host will be able to scrub through the current song playing, skip or change the song, adjust the volume and these changes will be reflected to the listeners players as well.

**Primary Actor:** User

**Precondition:** User must be connected in an ongoing listening session.

**Minimal Guarantees:** Changes are reflected to the listeners moments after it was made.

**Success Guarantees:** Changes are reflected to the listeners almost instantaneously after it was made.

**Trigger:** Host changes the media settings on their side.

**Main Success Scenario:** Changes are reflected to the listeners almost instantaneously after it was made.

**Extensions:**

* Connection of the host is lost when making changes.
* Media controls are made over a different device.

**ID: UC\_05**

**Title:** Control over session.

**Description:** Host will be able to manage who are in the listening session. And is able to end session whenever they want.

**Primary Actor:** User

**Precondition:** User must be connected in an ongoing listening session.

**Minimal Guarantees:** The host will be able to see the list of current listeners.

**Success Guarantees:** The host will be able to see the list of current listeners and are able to ban/kick them.

**Trigger:** Clicking on the options tab within the app.

**Main Success Scenario:** The host will be able to see the list of current listeners and are able to ban/kick them. The host will also be able to end the session whenever they please.

**Extensions:**

* None

**ID: UC\_06**

**Title:** View upcoming tracks

**Description:** User will be able to toggle a list of songs that are placed in queue.

**Primary Actor:** User

**Precondition:** User must be connected in an ongoing listening session.

**Minimal Guarantees:** The queue is shown to the end user.

**Success Guarantees:** The queue is shown to the users and are able to update within seconds of any changes.

**Trigger:** Clicking on the queue button within the app.

**Main Success Scenario:** The queue is shown to the users and are able to update within seconds of any changes.

**Extensions:**

* Preview of songs that are not available to that region.
* Preview of songs that are recently removed from queue.

**ID: UC\_07**

**Title:** Request songs to be put in queue.

**Description:** Listeners are able to put requests of songs they want to be added on the ongoing queue.

**Primary Actor:** User

**Precondition:** User must be connected in an ongoing listening session.

**Minimal Guarantees:** The request made by the listener will be sent to the database.

**Success Guarantees:** The request made by the listener will be sent to the database and changes are reflected on the queue immediately after the request has been made.

**Trigger:** Clicking on the request button within the app.

**Main Success Scenario:** The request made by the listener will be sent to the database and changes are reflected on the queue immediately after the request has been made.

**Extensions:**

* Requesting a song that isn’t available to the host’s region.
* Requesting a song when the request queue is full.