

Sprint 4 Plan

Spotify Tags

Sprint Completion Date: Tuesday, 14 March 2023

Goal(s):

- Add tags to entire albums or artists to expedite process.
- Load my information when app opens.
- To have a progress bar to be able to see song progression.
- To have volume controls when listening to song.

Task listing, organized by user story:

- **(5) As a listener, I want to be able to add tags to an entire album or artist.**
 - Create a React component for selecting a tag to add to songs. (3 hours) [Ryan]
 - Create a function to add a tag to a song from Spotify or from the library. (2 hours) [Victor]
 - Total for user story: 5 hours
- **(13) As a listener, I want my information to be loaded when I open the app.**
 - Create database update method/change insert to update too. (4 hours) [Dustin]
 - Put real data into the database. (4 hours) [Brandon and Dustin]
 - On login, load the user's library from the database. (3 hours) [Brandon]
 - api call to get each song stored for user
 - will end up using victor's userID state
 - or he will hook it up to your stuff
 - Write functions to wrap our backend calls. (3 hours) [Brandon]
 - Turn userID into state. (2 hours) [Victor]
 - use spotifyAPI call to get (see logout method in Home.jsx)
 - used for login/logout/getAllTags
 - Total for user story: 12 hours
- **(21) As a listener, I want to be able to add and delete tags from songs so that I can customize the tags on each song.**

- Add tags to a song when viewing its details. (6 hours) [Ryan]
 - Shows the component of list of tags to choose from
 - If type one that doesn't exist, have button to add it
 - creates tag and adds it to that song
 - Add tags to the expression. (6 hours) [Ryan]
 - Shows the component of list of tags to choose from
 - Total for user story: 12 hours
- **(5) As a listener, I want to have labels for what things do in the frontend.**
 - Create HTML titles for clarity on clickable elements. (4 hours) [Diana]
 - creative liberty for what they say, but here's a list of minimum things to add titles to
 - left-half of song in library
 - plays song
 - right-half of song in library
 - opens song card
 - login button
 - logout button
 - refresh list
 - not even sure if this is a needed component at this point
 - bottom half of SongCard ('bottom-half' element)
 - Song in LibraryResults
 - Song in SpotifyResults
 - On tags in the Expression or SongCard
 - deletes them
 - Remove URIs as titles of <tr>s representing songs. (2 hours) [Diana]
 - Total for user story: 6 hours
 - **(5) As a user I only want songs that I added tags to to show up in my library.**
 - Adding a tag to a song in SpotifyResults adds it to the library (and database). (3 hours) [Ryan]
 - Removing all of a song's tags removes it from the library. (2 hours) [Ryan]
 - Total for user story: 5 hours
 - **(5) As a user I want the currently shown library to reflect the list of songs that are playing.**
 - Check for if someone changes the library/updatedLib while playing a song. (3 hours) [Armaan]

- do something appropriate
 - new call to player?
 - Will that stop a song mid playing?
 - maybe it does nothing??
 - not sure if its too weird or unintuitive how it works now, haven't checked
 - probably, check after every song finishes (player changeState),
 - Total for user story: 3 hours
- **(13) As a user, I want to see how far I am through the song.**
 - Create a react component to show the song progress. (3 hours) [Armaan]
 - Be able to scrub through the song progress bar to change the song position. (5 hours) [Armaan]
 - way less important, not a big deal if not implemented
 - Total for user story: 8 hours
- **(3) As a listener, I want to be able to see and control the volume of the music playing.**
 - Display volume on screen (1 hour) [Diana]
 - Create React up & down button components (1 hour) [Diana]
 - Use material UI to style the up & down buttons (1 hour) [Diana]
 - Hook the buttons up to lower & raise the volume (1 hour) [Diana]
 - Total for user story: 4 hours

Extra:

- Refresh the accessToken Spotify API every call. (2 hours) [Ryan]

remove all errors (when logged out)

change volume buttons to slider: <https://mui.com/material-ui/react-slider/>

Team Roles:

- Ryan Gragg: Product Owner, Developer

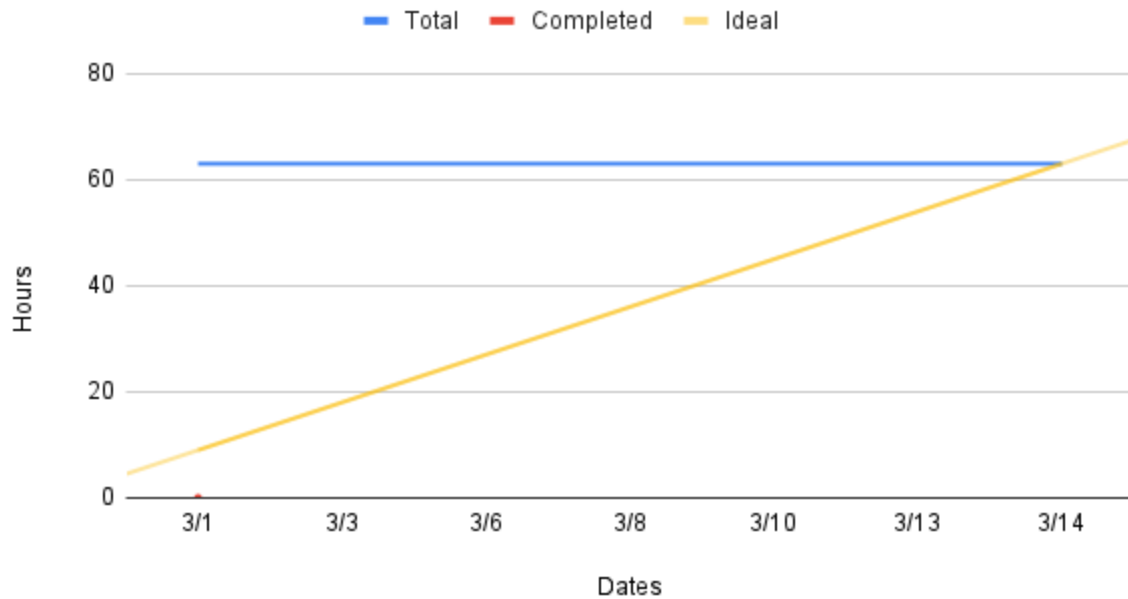
- Brandon Shien: Developer
- Diana Flores: Developer
- Armaan Deol: Scrum Master, Developer
- Dustin Creech: Developer
- Victor Derbier: Developer

Initial task assignments:

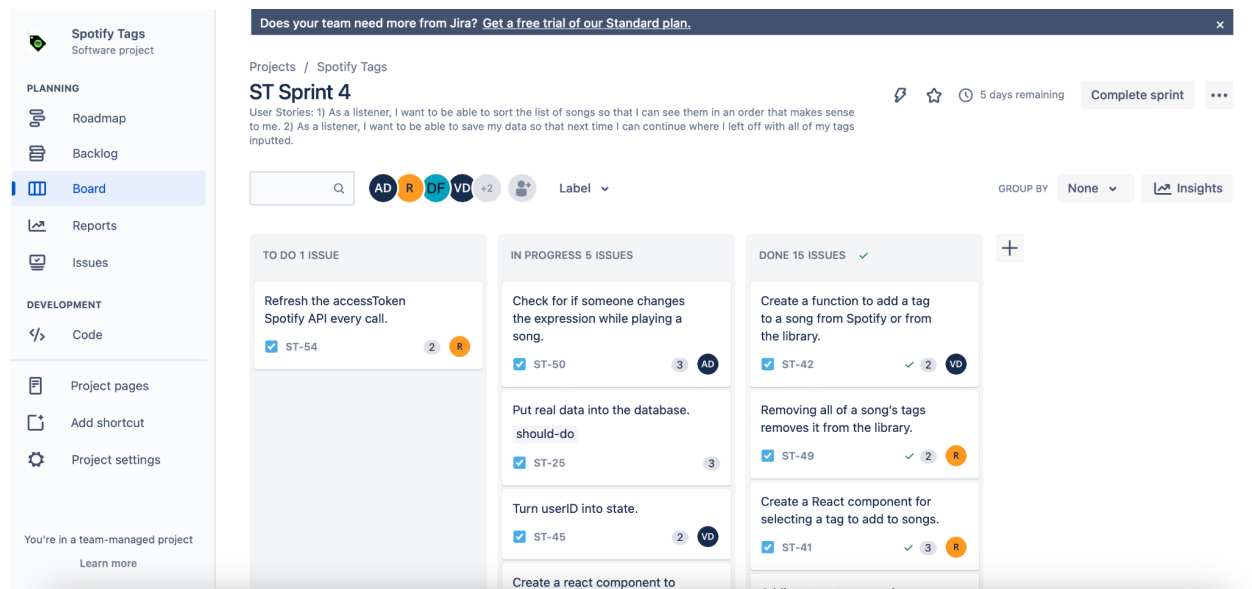
- Ryan Gragg: As a user I only want songs that I added tags to to show up in my library, so that I know what songs are actually in my library. As a listener, I want to be able to add and delete tags from songs so that I can customize the tags on each song, so that the songs in my library are fully customized.
- Brandon Shien: As a listener, I want my information to be loaded when I open the app, so that I can see all my information from Spotify without having to recreate it.
- Diana Flores: As a listener, I want to be able to control the volume, so that I can change it to my preference. As a user, I want to have labels in the frontend, so that I know what everything does.
- Armaan Deol: As a user, I want to see how far I am through the song, so that I can skip to different parts of the song. As a user I want the currently shown library to reflect the list of songs that are playing, so that when I update the playlist I hear my new songs.
- Dustin Creech: As a listener, I want my information to be loaded when I open the app, so that I can see all my information from Spotify without having to recreate it.
- Victor Derbier: As a listener, I want to be able to add tags to an entire album or artist, so that I can expedite how fast tags could be added and don't have to go one by one.

Initial burnup chart (include both initial & final burnup chart in Report):

Sprint 4 Burn-up



Initial scrum board:



Scrum times:

- “Daily” Scrum Meetings: **Monday, Wednesday, and Friday from 3 - 3:15 pm**
- Weekly Scrum Meeting with TA: **Friday from 3:15 - 4 pm**

Definition of Done (DoD)

1. For User Stories

- all DOD for tasks are completed

2. For Tasks

- code passes React.js linting
- documentation for each function is completed
- passes review(s) from appropriate reviewer(s) before merging into master branch
- new feature is fully merged and integrated
- Product Owner reviews added features