Sprint 3 Plan
Spotify Tags
Sprint Completion Date: Tuesday, 28 February 2023

Goal(s):

- To use buttons to go to previous or next song in list of songs.

- Be able to add tags to songs to create a list of songs. With these tags the user will be able to add/delete tags as well as be able to save the tags to use for another time

- To see a list of songs that depends on the given tag expression

- Create a database and connect that database to the backend

Task listing, organized by user story:

- **(3) As a listener, I want to be able to skip songs, so that I can skip over songs I don't want to listen to at the moment.**

  - Create React skip/prev button component (1 hour) [Diana]

  - When playing song, give list of songs instead of just one (1 hour) [Diana]

  - Use material UI to style the play, skip, prev buttons (1 hour) [Armaan]

  - Hook the buttons up to skip/prev the current song (1 hour) [Armaan]

  - Total for user story: 4 hours

- **(3) As a listener, I want to be able to have automatically added tags like artist and album, so that I can automatically listen to songs of the same artist or from the same album without adding tags to each song.**

  - Use Spotify API to get the artist and album information to make special tags (that are not removable). (3 hours) [Diana]
    - make method to get all artists and album names
      - expensive??
      - going to be used for seeing list of tags to add to expression
    - modify Victor's validateExpression and parseExpression to work with artist and album names??
    - Used when getting the list of tags to use in expressions
    - **NOT** actual tags for the songs, just make them act like tags

- ■ Related to Ryan's task of building wrappers for the tags viewer

  - ○ Total for user story: 3 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
      - ● show list of boolean operators
      - ● show name of all artists and albums in library

  - ○ Create a component to display the songs in your library. (5 hours) [Armaan]
    - ■ shows up when click on search bar, disappear when click off or click on song
    - ■ automatically updates as you type
    - ■ searches for songs in spotify, displays song img, artist, album
    - ■ wireframe pg 1

  - ○ Total for user story: 17 hours

- ● **(13) As a listener, I want to be able to enter an expression that changes the available list of songs so that I can only view the songs I want to at a certain time.**

  - ○ Construct the statement (tags picker) (8 hours) [Victor]
    - ■ shows a list of all tags that are on songs
      - ● does not include artist and album
      - ● should just be songs from get method implemented by Dustin
    - ■ wireframe pg 2 (Sprint 2 Mockups.pdf)
    - ■ Popover (see Songcard.jsx) for example
      - ● with **its own** search bar (steal from Armaan)
      - ● displaying songs is similar to SpotifyResults.jsx, showing a list of things that you can scroll through
        - ○ I think it should be fine to still make it a <table> even if there's just one row (row for tag)
    - ■ If you click on a tag
      - ● it adds that tag to the expression (and keeps the tags picker open)

- ○ and clears the tag search bar
  - ■ If you press enter
    - ● if there are any tags matching the search, it picks the first one (and keeps the tags picker open)
      - ○ and clears the tag search bar
  - ■ If you press escape
    - ● it closes the tags picker

- ○ Total for user story: 8 hours

- **(8) As a listener, I want to be able to sort the list of songs so that I can see them in an order that makes sense to me.**

  - ○ Create a front end interface to choose a method of sorting. (7 hours) [Brandon]
    - ■ I think pretty much everything will be alphabetical sorting either A-Z or Z-A
    - ■ maybe pick which attributes to sort by
      - ● primary sort, secondary sort, etc
      - ● based on song name, artist name, album name, number of tags?, (even info not displayed like release time?)

  - ○ Use JS to sort the list of displayed songs based on the sorting described. (5 hours) [Dustin]
    - ■ use js Array.prototype.sort, passing it a function
    - ■ the function compares two songs but also depends on the given sorting statement
      - ● unsure of the details of

  - ○ Total for user story: 12 hours

- **(3) As a listener, I want to be able to display the currently playing (or paused) song.**
  - ○ Create extra state to track currently playing song, update state when necessary. (2 hours) [Ryan]

  - ○ Highlight the currently playing song using CSS or React styling. (1 hour) [Ryan]

  - ○ Total for user story: 3 hours

Team Roles:

- Ryan Gragg: Product Owner, Developer

- Brandon Shien: Developer

- Diana Flores: Scrum Master, Developer

- Armaan Deol: Developer

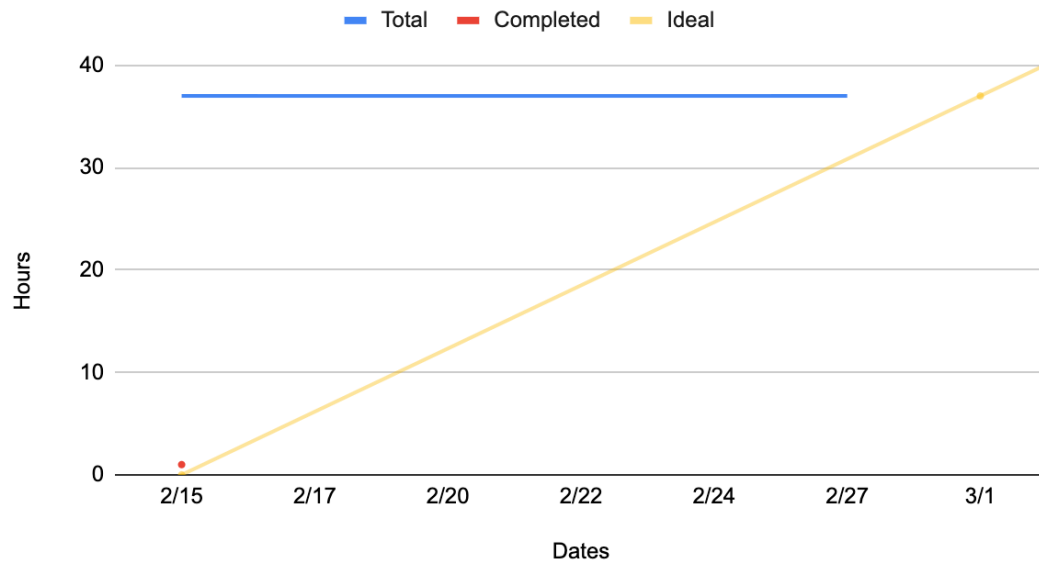- Dustin Creech: Developer

- Victor Derbier: Developer

Initial task assignments:

- Ryan Gragg: 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.

- Brandon Shien: As a listener, I want to be able to sort the list of songs so that I can see them in an order that makes sense to me. Create a front end interface to choose a method of sorting.

- Diana Flores: As a listener, I want to be able to skip songs, so that I can skip over songs I don't want to listen to at the moment. Create React skip/prev button component. When playing song, give list of songs instead of just one. As a listener, I want to be able to have automatically added tags like artist and album, so that I can automatically listen to songs of the same artist or from the same album without adding tags to each song. Use Spotify API to get the artist and album information to make special tags (that are not removable).

- Armaan Deol: As a listener, I want to be able to skip songs, so that I can skip over songs I don't want to listen to at the moment. Use material UI to style the play, skip, prev buttons. Hook the buttons up to skip/prev the current song. 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. Create a component to display the songs in your library.

- Dustin Creech: As a listener, I want to be able to sort the list of songs so that I can see them in an order that makes sense to me. Use JS to sort the list of displayed songs based on the sorting described.

- Victor Derbier: As a listener, I want to be able to enter an expression that changes the available list of songs so that I can only view the songs I want to at a certain time. Construct the statement (tags picker).

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

Sprint 3 Burn-up

Legend: ▬ Total   ▬ Completed   ▬ Ideal

X-axis: Dates (2/15, 2/17, 2/20, 2/22, 2/24, 2/27, 3/1)
Y-axis: Hours (0, 10, 20, 30, 40)

-

Initial scrum board:

Spotify Tags
Software project

PLANNING

Roadmap

Backlog

Board

Reports

Issues

DEVELOPMENT

Code

Project pages

Add shortcut

Project settings

You're in a team-managed project
Learn more

ST Sprint 3

User Stories: 1) As a listener, I want to be able to skip songs, so that I can skip over songs I don't want to listen to at the moment. 2) As a listener, I want to be able to queue songs, so that I can temporarily listen to songs out of order before returning to that order. 3) As a listener, I want to be able to have automatically added tags like artist and album, so that I can automatically listen to songs of the same artist or from the same album without adding tags to each song.

DF AD DC VD R

Label

TO DO 6 ISSUES	IN PROGRESS 2 ISSUES	DONE 2 ISSUES ✓
Use material UI to style the play, skip, prev buttons ☑ ST-37 — 1 AD	When playing song, give list of songs instead of just one will-do ☑ ST-17 — 3 DF	Create a server to access the database from backend. ☑ ST-11 ✓ 6 DC
Hook the button up to skip/prev the current song should-do ☑ ST-19 — 2 AD	Create a component to display the songs in your library (fitting the searchbar) ☑ ST-14 — 5 AD	Create React skip/prev button component should-do ☑ ST-18 ✓ 1 DF
Implement a queue of songs to be played before continuing with the list. When going to the next song, take from the queue if there are any songs there. should-do		

ST Sprint 3

User Stories: 1) As a listener, I want to be able to skip songs, so that I can skip over songs I don't want to listen to at the moment. 2) As a listener, I want to be able to queue songs, so that I can temporarily listen to songs out of order before returning to that order. 3) As a listener, I want to be able to have automatically added tags like artist and album, so that I can automatically listen to songs of the same artist or from the same album without adding tags to each song.

DF AD DC VD R

Label

TO DO 6 ISSUES	IN PROGRESS 2 ISSUES	DONE 2 ISSUES ✓
☑ ST-20 — 5		
Make artist and album 'tags' used in expression building should-do ☑ ST-21 — 3 DF		
Construct the statement ☑ ST-15 — 8 VD		
Add tags to a song when viewing its details ☑ ST-31 — 6 R		

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