Release Plan

**Product Name**: Group-ify

**Member Names**: Ashkan Shams, Hannah Wong, Kaitlyn Liao, Spencer Fulgham, Zachary Miller

# High Level Goals

Our main mission is to make listening to music on Spotify a more social experience. We will make a website that makes sharing your music tastes much easier to do. By logging in to our website using your Spotify account, users can customize their profiles as well as view their friends’ profiles to discover new music and stay connected.

# User Stories

### Sprint 1 - due Tues, April 20

research and planning will be done at this point

### Sprint 2 - due Tues, May 4

“As a Spotify user, I want to be able to login using my spotify account and have access to my profile.”

Story Point Estimate: 13

"As a Spotify user, I want to be able to visually change my profile so that it matches along to the aesthetic of my music tastes."

Story Point Estimate: 21 (or like an 13 idk)

"As a Spotify user, I want to be able to see my top listened to tracks so that I can gain insight on my own music taste."

Story Point Estimate: 13

### Sprint 3 - due Tues, May 18

“As a Spotify user, I want to be able to view my friend’s profile so that I can see their music taste.”

Story Point Estimate: 21

“As a Spotify user, I want to be able to easily search for songs and albums so that I can add them to a list with music I’d recommend to my friends.”

Story Point Estimate: 21

### Sprint 4 - due Tues, June 1

"As a spotify listener, I want a way to listen to along to music with others concurrently so that we can both enjoy a song without being together in a room"

Story Point Estimate: 34

“As a friend group, we want to be able to debate which songs to listen to together so that the whole group is happy with the music choices”

Story Point Estimate: 34

# Current Release Features

1. Login and connect your personal Spotify account
   1. Difficulty: 2
2. See and display your analytics throughout the year.
   1. Difficulty: 5
3. Change the color palette of the browser to reflect certain music types
   1. Difficulty: 13
4. Interactive image of song/album covers to display your top tracks or albums
   1. Difficulty: Really depends, might be 21 or it might be 34
5. See your friends on browser so you can view their profile
   1. Difficulty: 21
6. Invite active friends to a group listening party
   1. Involves Synchronize the experience between multiple people
   2. Difficulty: 34
7. upvote and downvote songs in group listen queue
   1. Difficulty: 8

### Potential Problems

1. Rate Limiting: enables Web API to share access bandwidth to its resources equally across all users.
2. Spotify has some weird rules, one of them being when and how to use the Spotify logo.
3. Are we allowed to store user information in databases?
4. It appears that Spotify doesn’t make finding friends easy. There is a github that might make getting friends easier. If not, we might need a login system and have users manually add their friends to our website.
5. Privacy issues with other users to see your analytics without being logged into your account.

# Product Backlog

1. Rate songs. Rate your friend’s favorite songs. Let people rate your favorite songs. Display your ratings and see what people rated your songs.
   1. Will definitely need work with databases, might have to store each user in a database. Then we need to store their favorite songs. Then we need to store the rating for each song, both the user’s opinion and other opinions.
   2. Needs to be turned off if necessary.
   3. Privacy concerns?
   4. Can we store people’s information or is that violating Spotify TOS?
   5. Should we have an explore feature? Where you click an explore button and you are sent to playlists of random users and asked to rate them?
   6. Implement a way for users to input a score for a song. Store Point Estimate: 1
   7. Store rating in a database: Store Point Estimate: 1
   8. Implement a way for users to see their ratings of songs and change their scores. Will need a separate page on the website but we might find making a new page easy by then. Store Point Estimate: 2
   9. When a user searches for a song they should be able to see what their friends rated it. Store Point Estimate: 2
   10. Difficulty: Lots of concerns, will involve the most unique work rather than messing around with the Spotify API. Without the explore option and if we just deal with you and your friends it might be 13 or 21.
   11. With the explore option 34.
   12. Spikes
       1. Spike 1: Part of Sprint 2, Learn how to search for songs/albums so that users can assign a rating to them.
       2. Spike 2: Learn the optimal way for rated songs to be stored per user. Meaning the way that reduces the size of our tables the most.
       3. Spike 3: Learn how users can see how their friends rated a song.
       4. Spike 4: Learn how to add a song that is rated to a database.
2. Playlist sorting by popularity or score
   1. Very low priority
   2. It seems like there is a way to modify playlists, but this might permanently change the order rather than sort them.
   3. Collaborative playlist where friends can suggest and vote of what songs get added to it Story Point Estimate: 13 (Hopefully this will come down with research)
   4. Difficulty: 2
3. See shared liked songs between friends
   1. Low priority
   2. Difficulty: 3 (This will likely involve databases).
4. Store listening history or friend activity
   1. Low priority
   2. Difficulty: 3 (This will likely involve databases).

# Sprint Planning

## Sprint 1:

Goal: Our goal is to familiarize ourselves with the technologies. This research will allow us to develop a basic foundation for our web application.

* Spike 1: Brush up on:
  + CSS we will use (Bulma)
  + Multi-User GitHub
  + Py4web
* Spike 2: Learn Spotify API
  + Learn how tokens work in the Spotify API. Like do we need to get a token for each user that uses the site? What happens if we don’t get one or it fails?
  + Learn py4web database management and how we can implement Spotify API into it.
  + Learn how to get and interact with user analytics
* Spike 3: Learn how to make a separate pages user profiles (including setting page)
* Spike 4: Learn how to make/fit different tiles for each feature
* Spike 5: Learn how to store information per user in a py4web database, (needed for top and recommended tracks/albums.)
* Spike 6: Learn about search bar implementation

### Tasks:

1. Make a github for code, and maybe a separate google drive folder for resources and sketches.
   1. Story Point Estimate: 1
2. We need to make a skeleton website index at least. It doesn’t need to be pretty.
   1. Story Point Estimate: 1
3. Make logging in with Spotify possible on our website.
   1. Story Point Estimate: 3
4. Make users change their theme or color. In this case it might just be for the site. If we want themes/colors to change only for user profiles, this should be moved to sprint 2 and we should do analytics first.
   1. Story Point Estimate: 3
5. (Analytics) Have a section on the website a user can click to see their top tracks of all time. We just need names if getting pictures of tracks is too difficult for now. See resources linked at the top for this.
   1. Story Point Estimate: 3 (May have to be finished in sprint 2)

## Sprint 2

Goal: We want to focus on developing the user profiles features and capabilities for customization, while tackling our spikes regarding a search bar and friend lists.

* Spike 7: Learn how multiple users can login to our website.
* Spike 8: Learn how users can search for albums/songs/playlists and add/display it on their profile.
  + Is bulma CSS enough?
  + If not, what can help us?
  + How difficult is it to implement a GUI for the user.
  + How structured do we need the profile to be? Should we
  + limit the template to basically just a set number of squares
  + a user can fill in?
* Spike 9: Learn how we can make a user see their spotify friends and manipulate/interact with your friends list on spotify:
  + we need to add their friends to a database?
  + This may be hard if the API does not support this. We might have to make users manually add their friends.
  + Can we accomplish this by sharing a link between users?
* Spike 10: Learn how to control privacy of your profile
  + Learn how users can see their friends analytics
  + Learn how users can see their friends top/recommended songs.
  + Learn how users can see their friends' public playlists.

### Tasks:

1. Make a user be able to click somewhere to access their personal page.
   1. Story Point Estimate: 1
2. Finalize profile/settings page UI
   1. Story Point Estimate: 2
3. Allow a user to put their playlists inside our profile structure. Store their profile in the database.
   1. Story Point Estimate: 3
4. Have a section to display user’s spotify analytics. We just need names if getting pictures of tracks is too difficult for now. See resources linked at the top for this.
   1. Story Point Estimate: 3
5. Start Implementing a search function so that users can find the albums/tracks they want to display.
   1. Story Point Estimate: 5

## Sprint 3:

Goal: We want to develop the features that allow users to connect and view friends’ profiles, as well as allowing users to control the privacy and visibility of features on their own profiles.

* Spike 11: Learn how to access a user’s currently listened to song.
* Spike 12: Investigate how discord is able to accomplish shared listening.
* Spike 13: Investigate if we can queue songs for users, (if it is not possible then the best thing we can do is make a playlist of songs rather than a queue).
* Spike 14: Investigate if any other platform has implemented queued spotify listening.

### Tasks:

1. Finish search bar implementation so that users can find the albums/tracks they want to rate. (Continued from previous sprint)
   1. Story Point Estimate: 3
2. Implement a way for a user to click on their friends’ page and see their profile.
   1. Privacy settings
   2. Analytics privacy
   3. See their top/recommended songs
   4. See public playlists
   5. Story Point Estimate: 3
3. display friends list
   1. be able to navigate to their profile (if they have one)
   2. Display favorite song (similar to discord status)
4. Start group listening implementation

## Sprint 4:

Goal: Implement a group listening feature as well as generally cleaning up the web app ad tying loose ends.

----------------------------------------------------------------------------------------------------

Note: If we are behind in our sprints then this will be used to catch up on anything we missed the deadline for.

----------------------------------------------------------------------------------------------------

### Tasks:

1. Implement shared listening. (Continued from previous sprint) Story Point Estimate: 13
   1. implement an easy way to start/invite friends
   2. Upvote and downvote songs in queue
2. Clean up the website. (This will be done throughout the sprints but this will be the finishing touches) Story Point Estimate: 1
3. Extensive testing and finish resolving existing bugs. (This will be done throughout the sprints but this will be the finishing touches) Story Point Estimate: 1

# [Initial Project Presentation](https://docs.google.com/presentation/d/1wG_CsESkz8U_tEP_9At_rjrYsJEeCCpsLi8NwtyI3QY/edit#slide=id.gcf50ac6991_0_67)