

**Team 23 - Product Backlog**

Ronak Bharad, Donald Bough, Ivan Chan, Kareem Elhadidi, Siddharth Patel, Timothy Poozhikala

**Problem Statement**

Playing the wrong music at a party can completely ruin the experience for the guests and managing the music can be tedious for the hosts. Hiring a DJ could solve this problem, but cost is often prohibitive. Our goal is to make music selection accessible to our guests and carefree for our hosts. Our webapp automatically suggests songs based on the music tastes of the guests and allows them to vote on what they want to hear.

**Background Information**

**Users**

Our target audience consists of anybody who attends or hosts events ranging from a group of friends to large private events. Fed up of people taking over your phone and music systems at the parties to play their songs instead? Our webapp allows the guests to vote for their favorite songs and decide which ones should play next. Now, hosts can enjoy the party too.

**Similar Applications**

There are a few similar applications such as Festify, Jukestar, and Populrr that also have the voting feature. Festify allows people to vote for songs to push them up the queue, while Jukestar makes it so that you can veto songs off the list if enough people vote.

**Limitations**

Festify offers guest voting but none of the applications will automatically keep the party going by queuing songs based on guests’ tastes. There are several music applications such as Festify, Jukebox, Trackt, etc. that have the voting feature to play specific songs. However, they don’t have a continuous list of suggestions based on the current taste of the guests. Our webapp will use the preferences of the guests that are currently present, which none of the other applications can do. There is one application with seemingly similar features called Spartify, but upon testing it appeared to be nonfunctional.

**Requirements**

**Functional**

*As a GUEST I would like to:*

1. Join an event given a passcode
2. Vote for a song
3. See what other songs are being voted for
4. Look at songs similar to one playing
5. Give feedback about overall music selection to the host
6. Connect my spotify account
7. Browse my public spotify playlist
8. See what songs have already been played
9. Get recommendations based on my past votes
10. Keep my information anonymous from DJ and others
11. Keep my votes anonymous
12. See how popular a song is based on Spotify databases
13. Be notified when voting is opened
14. Have more voting power the more my songs are played
15. Add songs from the queue to my Spotify playlist
16. Get a notification when I am able to vote again
17. Pay to move my song closer to the top of the queue **(if time permits)**
18. Hear song previews **(if time permits)**
19. Get recommendations for myself based on time and location **(if time permits)**

*As a HOST I would like to:*

1. Generate an event passcode for my guests
2. Be able to look at a list of all played songs after the event
3. Create a playlist played songs after the event is over
4. Create a fallback playlist in case there isn’t enough request
5. Receive feedback about overall music selection from the audience
6. Close and open voting
7. Users to be limited on the number of votes they have
8. Give users more votes or power to vote
9. Analyze what songs have been played at my event location before **(if time permits)**
10. Look at what songs are popular in my town/city**(if time permits)**
11. Have “recommendation” mode where I only see the songs and they don’t actually play **(if time permits)**

**Nonfunctional**

*As a GUEST I would like:*

1. A simple to navigate interface
2. Easily connect to a group event
3. To use less data while using the web app
4. As a user I would like to use GPS for getting recommendations **(if time permits)**

*As a HOST I would like:*

1. A simple to navigate interface
2. To easily start an event
3. To not have my device overwhelmed by too many guests.

*As a DEVELOPER I would like:*

1. Our app to be able to work on different browsers.
2. To integrate nicely with backend
3. To use efficient algorithms
4. To implement simple machine learning libraries/sample code.
5. To have a secure connections from clients to host
6. To have databases that can be easily scalable
7. To see our web app scale correctly on different sized devices (e.g. Tablets, laptops, phones) **(if time permits)**