Bowdoin College

Computer Science Department

CSCI 3715

DJ Hero: Creating and Evaluating a System for Sharing Music Preferences

Authors:
Dylan Hayton-Ruffner, *@hrdylan*
Ian Stewart, *@ipstwart*
George Benz, *@gbenznyc*

May 10, 2020

1 Abstract

This study explores the design, development, and testing of a novel user interface for the web app DJ Hero. We take an html/css/js frontend from wire frame to high fidelity prototype, and evaluate the effectiveness of the design against our competitor Spotify. Participants in the study were asked to complete similar tasks in both UIs, and provide qualitative feed back on the aesthetics of both designs. We were unable to reject the null hypothesis and could make no definitive claim about the efficiencies of the designs of Spotify and DJ Hero. Our study indicated users had difficulty bridging the gulf of execution in the DJ Hero UI. We discuss further iterations of DJ Hero and future studies of other aspects of the system's functionality.

2 Introduction

Music plays a part in every social gathering. Whether it is in the background at a dinner party or at the forefront during a dance party, music provides a tool for the host to build an inclusive atmosphere for their guests. With something as diverse as music, how can a host curate music to be as inclusive as possible?

After surveying many of our friends at Bowdoin and reading critques in the Bowdoin Orient [1], we found that music was a divisive motivator in whether a student would feel included and welcome in a given social gathering. Our research mirrored our own perceptions of music at social gatherings as both hosts and guests. The music was an extremely delicate and important part of a social gathering but was often not given the time and attention needed to be as inclusive as possible.

We seek to optimize the creation and curation of playlists by creating a platform for Bowdoin students to share and discover what other students are listening to on campus.

In this paper we will first go into further detail regarding the motivation and research behind our topic. We will talk about the details of existing solutions that our solution attempts to build on. In the approach section we will then specify our attempts that build on previous solutions. In the methods section we will describe the summative evaluation we have pursued for our project. Next in the results section we will present our findings. Finally in the discussions section we will look back, look forward, and conclude the work overall.

3 Related Work

In this section we look at possible competitors to our solution. We will look at the positive design aspects and any key problems or limitations so we can better understand where our product can fit in the marketplace.

[1]https://bowdoinorient.com/2017/04/26/youre-not-that-woke-on-the-importance-of-music-culture-and-self-doubt/

3.1 Overview of Competitors

While conducting research on our competitors we looked at 6 different solutions, Festify, Soundshare, Jukestar, Mubo, Troppo and finally Spotify. Most of our research efforts focused in on Spotfiy as its solution was most popular at Bowdoin.

Festify:[2] Festify works by creating a party in your broweser that guests can join. Then from that party guests can request and vote for songs that they want to hear the most.
Soundshare:[3] Soundshare is a social music networking app that allows its users to see what their friends are listenting to, what playlists they are making, no matter what music service they use.
Jukestar:[4] Jukestar works by automatically aggregating the votes of what songs guests want to listen to. It then uses this feedback to generate playlists that will play the songs with the most votes sooner and veto songs that people dislike.
Mubo:[5] Mubo works by creating a public "queue" which allows users to customize the queue through their votes and inputs. By using their smartphones they can decide how soon a song is played.
Troppo:[6] Troppo works by creating a "party" of its guests and hosts. Troppo then allows its users to vote and request songs which become visible to the host.

Competitive Solutions				
Solutions	Collaborative Playlists	Real Time Feedback	Compatible with Music Platforms	Auto-generates Playlists
Festify	No	No	Spotify	No
Soundshare	Yes	No	All	No
Jukestar	No	Yes	Spotify	Yes
Mubo	Yes	Yes	Spotify, Deezer	No
Troppo	No	Yes	Soundcloud, Youtube	Yes
Spotify	Yes	No	Spotify	No

Table 1: Compares the functionalities of each competitor.

[2]https://festify.rocks/faqs
[3]https://www.soundshareapp.com/
[4]https://jukestar.mobi/
[5]https://www.mubo-app.com/
[6]http://www.troppo.me/

3.2 User Study Competitor: Spotify[7]

While the other platforms we have looked at provide novel solution to many of the issues we proposed, many of them act as third-party services. Spotify allows users to stream a large library of available music via playlists that can be private, public, or collaborative. Although multiple users can contribute to a collaborative playlist, only the user playing the music can control the queue. Spotify has a huge advantage because it is widely used and has a massive network of users as an established social platform, but lacks any real-time feedback or collaboration when playing music.

3.3 Our Contribution

In conclusion, much of the features we plan to include in our solution are already found in other products online. However, no single solution included every feature that we plan to have in our final product. Another way to differentiate our application would be to add party music discovery features. Because parties across Bowdoin would be occurring on our platform, the app could leverage that data and show users what was hot on campus. This kind of information intrigues users and adds value beyond the initial scope of the application.

4 Approach

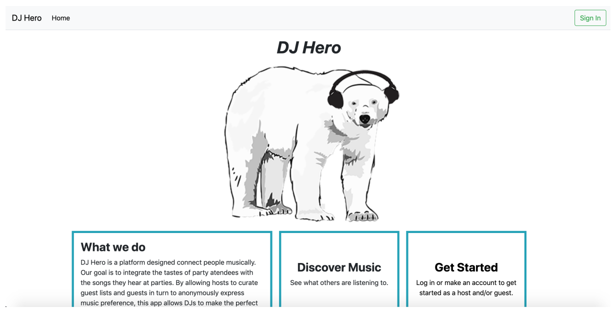
The DJ Hero UI was designed to support 4 core tasks in the DJ Hero application: account sign in/creation, user playlist creation, party playlist construction, and music discovery.

4.1 Account Sign-in/Creation

The DJ Hero UI features a basic account sign-in and creation work flow. Upon entering the home page, displayed in Figure 1, the sign-in button is shown in the top right hand corner. Both the placement and text of this signifier, match the standards of real-world web design. The user expects this button to be in the top right because most UIs place account information in the top right hand corner of the navbar. On hover, the button fills and on click it highlights. This helps bridge the gulf of evaluation by providing feedback to the user about the position and state of their mouse as they navigate the site. All buttons and clickable objects on the site conform to this standard.

After clicking on sign-in button, the user is greeted with the sign-in widget, shown in Figure 2. This widget, and the create account widget, focus on simple designs that conform to common standards. This has the potential to help the user bridge the gulf of execution because they have performed the task before in similar contexts. When conforming to standards, subtle details are vital.

[7]https://support.spotify.com/us/using$_{s}potify/playlists/create-playlists-with-your-friends/$



What we do

DJ Hero is a platform designed connect people musically. Our goal is to integrate the tastes of party atendees with the songs they hear at parties. By allowing hosts to curate guest lists and guests in turn to anonymously express music preference, this app allows DJs to make the perfect

Discover Music

See what others are listening to.

Get Started

Log in or make an account to get started as a host and/or guest.

Figure 1: A screen shot of DJ Hero's home page.

For instance, on the sign in widget, the 'create account' text is highlighted and placed below the login.

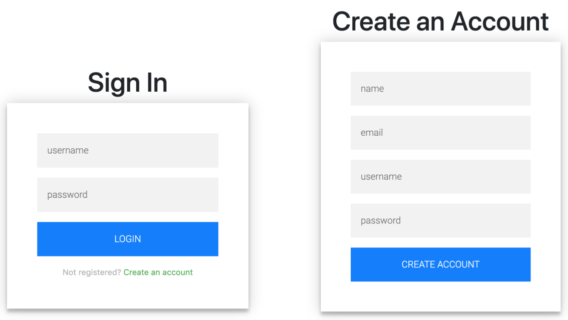


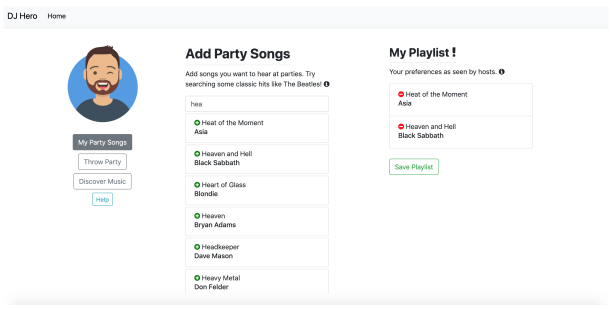
Figure 2: Screen shots of DJ Hero's sign-in and create account widgets.

4.2 User Playlist Creation

In this task the user selects songs that they like to hear at parties and adds them to their public-facing playlist. This page, displayed in Figure 3, is made up of two widgets - the adder and the editor.

The adder is on the left side of the page. Users search for songs and use the '+' icon to add them to their list. The search bar dynamically reacts to the user's query, which could help the user evaluate and plan their actions. Despite its clarity, one possible weakness of the adder widget is that no functionality is attached to the 'enter' key. In search bars, users typically use the 'enter' key to input their search. This gap between expectation and reality might cause difficulties for users as they plan their actions.

The editor widget, displays the user's current playlist. Songs are removed with the '-' icon. The most important function of this widget is that it displays user actions in real time. When a user adds or removes a song, the playlist is immediately updated to reflect this change in state. Since this update as instantaneous, the user likely perceives it as causally linked to their deletion or addition action. This has the potential to give the user a sense of control and empowerment. Immediate updates also increase the efficiency of the interface by allowing the user to detect mistakes in real time. The '+' and '-' icons along with their coloring, signify ways in which the user can correct their actions. Thus, mistakes can be detected and corrected quickly. The editor widget also displays the save state of the playlist through icons. When the playlist is not saved, an exclamation mark appears next to the playlist, indicating an important issue. Upon saving, the system communicates to the user that it is working through a spinner icon before showing success with a check mark. This flow is shown in Figure 4.



Add songs you want to hear at parties. Try searching some classic hits like The Beatles!

hea

Heat of the Moment
Asia

Heaven and Hell
Black Sabbath

Heart of Glass
Blondie

Heaven
Bryan Adams

Headkeeper
Dave Mason

Heavy Metal
Don Felder

Your preferences as seen by hosts.

Heat of the Moment
Asia

Heaven and Hell
Black Sabbath

Figure 3: Screen shot of the playlist page.

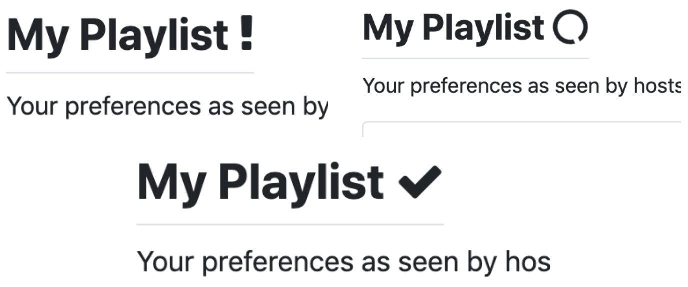

My Playlist !
Your preferences as seen by
My Playlist
Your preferences as seen by hosts
My Playlist ✔
Your preferences as seen by hos

Figure 4: Screen shots of the save playlist system response.

4.3 Party Playlist Creation

This task is the most complicated action in the DJ Hero application. The user must select guests, look at guest preference data, add songs and edit their playlist. To break down this difficult task, the page is divided into sub-tasks. This division decreases the complexity of the user's planning phase by simplifying the immediate task at hand. Each sub-task page is linked in a carousel, which communicates their linearity. Before diving into each sub-task, we discuss several widgets common to each of the sub-task UIs: the search widget, the navigation widget, and the editor widget.

4.4 Search Widget

The search widget operates differently than the search widget from the playlist page. As the user types text into the bar, the system searches its database. If a matching object is found, the bar highlights green and displays a check mark. Otherwise it highlights red. This flow is displayed in Figure 7. When a user inputs a valid song, they must hit 'return' to add the object to a list or playlist. This design decreases the complexity of the user's task, because the user is not required to navigate and select an icon in order to add an object. Thus, the task is in theory more efficient and much easier for users who have low dexterity. However, the widget has some potential draw backs. First, the colored system response is also commonly used in form completion to communicate the validity of input text. This may cloud the user's understanding of the task. In addition, the search function is binary. The system does not tolerate any mistakes the user makes when inputting the object. Thus, this widget trades conformity and tolerance for speed and mechanical simplicity. The widget is likely better suited to experienced users, who make fewer mistakes than newer ones. It is also potentially beneficial for disabled persons, who maybe find targeting small icons with the a mouse cumbersome and difficult.

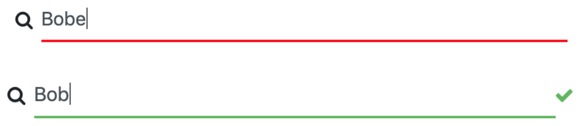


Figure 5: Search bar found response.

4.5 The Navigation Widget

The navigation widget is displayed at the bottom of every sub-task and allows the user to move between tasks. The widget has two buttons, 'back' and 'next' which, when clicked, move the user in the corresponding direction through the sub-tasks. Widgets at the ends of the carousel do not have the corresponding 'back' or 'next' button.

4.6 The Editor Widget

The editor widget is identical to the editor from the playlist page with one import distinction: its data is persistence across sub-tasks. Even if the user moves back in the task flow, the playlist state is preserved. Thus, the state of

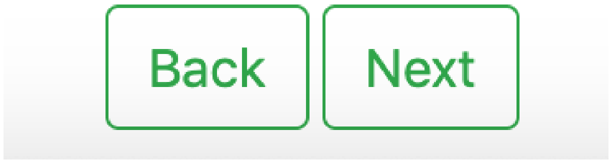


Figure 6: Screen shot of a navigation widget.

the overarching goal of the task - a party playlist - is always displayed to the user. This reminds them of the task's goal, and allows them to edit that goal at all steps in the task.

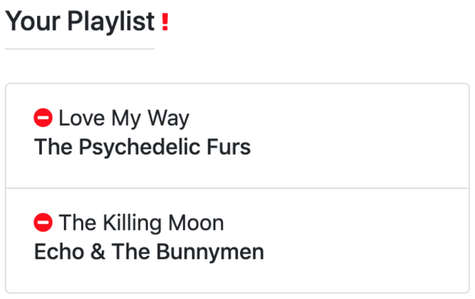


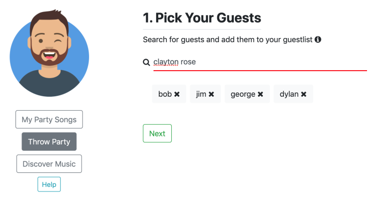
Figure 7: Screen shot of the editor widget.

4.6.1 Sub-Task: Add Guests

The add guests sub-task allows user's to add guests to their guest list. Guests found and added by users, appear beneath the bar and can be removed by clicking on the 'x' icon. The proximity of the list and search bar decrease the time required for a user to correct a mistake. It also articulates that they are related objects in the interface. Figure 8 shows the Add Guests sub-task.

4.6.2 Sub-Task: Add Guest's Favorite Songs

This page is very similar to the User Playlist Creation page. However, rather than present the user with a search bar, the adder widget displays their guest's favorite songs.



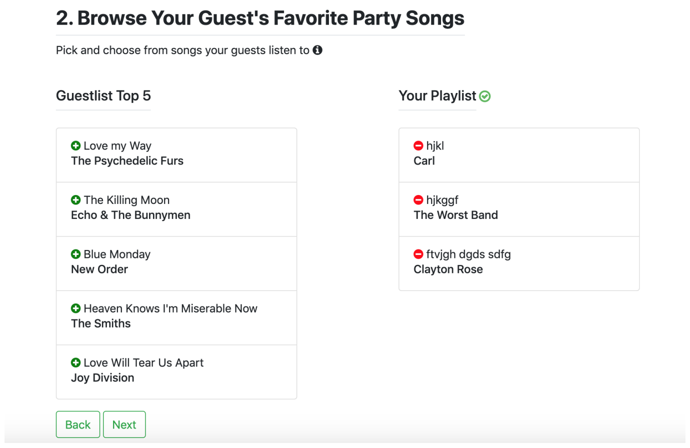
1. Pick Your Guests

Search for guests and add them to your guestlist

clayton rose

bob ✖ jim ✖ george ✖ dylan ✖

Figure 8: Screen shot of the add guests sub-task.



2. Browse Your Guest's Favorite Party Songs

Pick and choose from songs your guests listen to

Guestlist Top 5	Your Playlist
Love my Way The Psychedelic Furs	hjkl Carl
The Killing Moon Echo & The Bunnymen	hjkggf The Worst Band
Blue Monday New Order	ftvjgh dgds sdfg Clayton Rose
Heaven Knows I'm Miserable Now The Smiths	
Love Will Tear Us Apart Joy Division	

Figure 9: Screen shot of the Guest's Favorite Songs sub-task.

4.6.3 Sub-Task: Add Guest's Favorite Genres

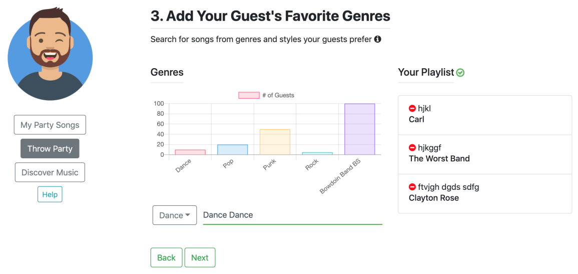
In this sub-task, users are presented with a graph indicating their guests genre preferences and a search function for adding songs by genre. The graph widget, quickly and clearly communicates guest preferences to the user. The search widget is identical to the widget discussed in section **??**

4.6.4 Sub-Task: Add Your Favorite Songs

In this sub-task, the user selects and adds their favorite songs using the search widget.

4.6.5 Sub-Task: Edit and Save the Playlist

In the edit and save sub-task, the user edits and saves their playlist. On save, a confetti animation is displayed to congratulate the user for finishing the task and to articulate that the task is complete.


3. Add Your Guest's Favorite Genres

Search for songs from genres and styles your guests prefer

Your Playlist

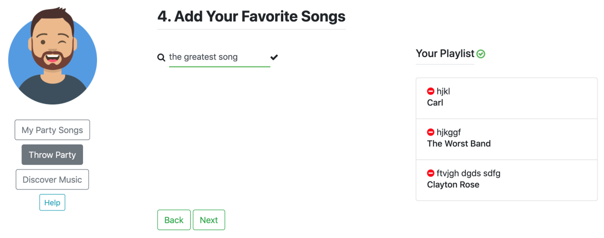
hjkl
Carl

hjkggf
The Worst Band

ftvjgh dgds sdfg
Clayton Rose

Dance Dance

Figure 10: Screen shot of the Add Guest's Favorite Genres sub-task.


4. Add Your Favorite Songs

the greatest song

Your Playlist

hjkl
Carl

hjkggf
The Worst Band

ftvjgh dgds sdfg
Clayton Rose

Figure 11: Screen shot of the add guests sub-task.

4.7 Music Discovery

The music discovery page aggregates all public user data and displays popular songs and playlists. The playlist widget allows the user to selectively view each playlist. This functionality prevents the user from being overwhelmed with data.

4.8 Help Documentation

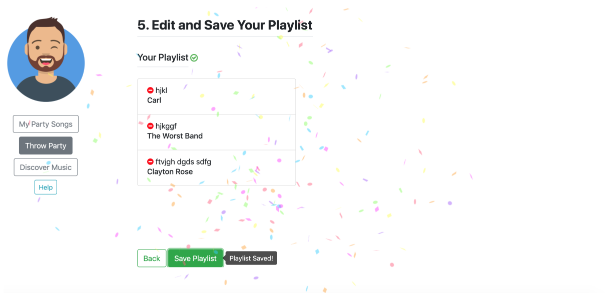
A blue help tab exists on all pages, to clarify the functionality of the UI and to assist users in completing their goals.

4.9 Formative Analysis: User Interviews

While developing the UI, we performed a series of user interviews, to gauge the effectiveness of our design. In these interviews, we asked users to complete each of the core tasks in the UI: account sign in/creation, user playlist creation, party playlist construction, and music discovery. These interviews led to the following findings.

4.9.1 Using Sub-Tasks in Party Playlist Construction

In our initial design, the Party Playlist Construction tab was not subdivided. Instead, all widgets were displayed on one page. This greatly confused our users.



5. Edit and Save Your Playlist

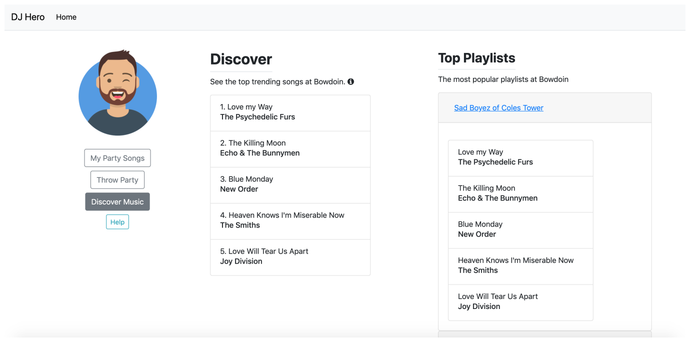
Your Playlist

hjkl
Carl

hjkggf
The Worst Band

ftvjgh dgds sdfg
Clayton Rose

Figure 12: Screen shot of the add guests sub-task.



Discover

See the top trending songs at Bowdoin.

1. Love my Way
The Psychedelic Furs

2. The Killing Moon
Echo & The Bunnymen

3. Blue Monday
New Order

4. Heaven Knows I'm Miserable Now
The Smiths

5. Love Will Tear Us Apart
Joy Division

Top Playlists

The most popular playlists at Bowdoin

Sad Boyez of Coles Tower

Love my Way
The Psychedelic Furs

The Killing Moon
Echo & The Bunnymen

Blue Monday
New Order

Heaven Knows I'm Miserable Now
The Smiths

Love Will Tear Us Apart
Joy Division

Figure 13: Screen shot of the Music Discovery page.

During our interviews, we noticed the users who were partially successful tried to complete the task in a step-by-step manner. This observation led us to the use of sub-tasks described above.

4.9.2 Abandoning 'Thrower' and 'Goer'

During app brainstorming, we used the terms thrower and goer to refer to users who attend parties and users who throw parties. Each of these users has different tasks in the interface. In our initial design, this nomenclature was used as part of the UI. This confused users, who had no idea what either term meant. Moreover, once the terms were explained to them, they conveyed that they would likely use both functionalities of the app. Thus, we decided to remove the notion of two personas from the user interface.

5 Methods

In conducting user experiments, we used an early iteration of our system and had participants test our interface for adding and changing song preferences. We compared our system with Spotify, a main competitor who has a large portion of the market share in the student demographic.

For our system, participants were asked to login from the homepage, and add a few songs of their choice from a pre-provided list on the User Playlist Creation page. Once they were satisfied with their choices, they were asked to save the playlist. For Spotify, users were instructed to open up the web or desktop app, create a playlist, add a few songs, then copy the link as if to send it to a friend. Both experiments tested the comparable task of aggregating music preferences to share. The experiment assumed users had a Spotify account, and participants were asked to reach out to us if they did not have an account so accommodations could be made. All of our participants had accounts.

In collecting participants for the study, we used a convenience sample. Participants were assigned to us and chosen from a pool of classmates in the same course that this project is for. We had 9 participants in total.

Experiments were conducted on the computers of the participants. Participants received instructions from a Slack message, and submitted responses through a Google Form. Over the course of each experiment, the participant visited our website and Spotify on internet browsers of their choice (or, in the case of Spotify, the desktop application). Users timed themselves as instructed using a personal timer.

The response form focused on time for task completion and qualitative descriptors of the two interfaces. Although it requested the names of participants, data was only looked at in aggregate form to ensure anonymity. Names were only requested so that individuals who did not complete the task by the deadline could be contacted. Participants reported time to complete task in seconds and then chose 3 terms that described the system from a provided list of 10 terms taken from the Microsoft Desirability Toolkit (See Appendix Figure A.3 for complete list of terms).

Our independent variables were the two systems, DJ Hero and Spotify. The dependent variables were the time to complete the tasks and descriptor words. The 9 participants were broken up into two groups, A and B. Group A (5 participants) first completed the task for our system, then Spotify. Group B (4 participants) first completed the task for Spotify, then our system.

Users participated individually, having been assigned the tasks via Slack. Under ideal conditions, participants would have been given the tasks in person and had researchers present to answer questions, but due to the quarantine associated with the COVID-19 pandemic, research was conducted remotely and participants were encouraged to ask any questions via Slack.

6 Results

The mean time to complete the task for DJ Hero was 79.72 seconds, with a standard deviation of 47.64 seconds. The mean time to complete the Spotify task was 53.91 seconds, with a standard deviation of 26.34 seconds. These results are illustrated in Figure 14.

Responses to the qualitative responses are shown in Figure 15. In regards

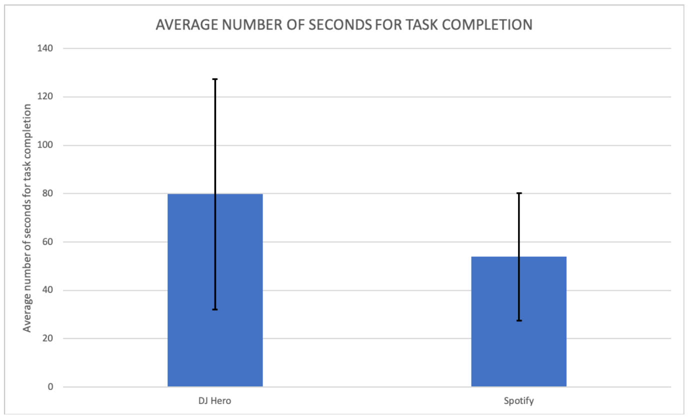


Figure 14: The average number of seconds it took for users to complete the assigned task for DJ Hero and Spotify.

to user interface, the descriptor chosen most for both our system and Spotify was "Attractive," with 7 choosing it to describe our system and 6 using it to describe Spotify. Similarly, our system did well for "Approachable," with 5 responses. Spotify generally scored higher when it came to terms regarding usability, scoring 5s for both "Intuitive" and "Convenient" and only a 2 for "Hard to Use" and 1 for "Confusing" whereas our system scored lower on "Intuitive" and "Convenient," with a 2 and 3 respectively and higher on "Hard to Use" and "Confusing," with a 4 and 2 respectively. Both systems got a 2 for "Satisfying," and no respondent thought either system was "Unattractive."

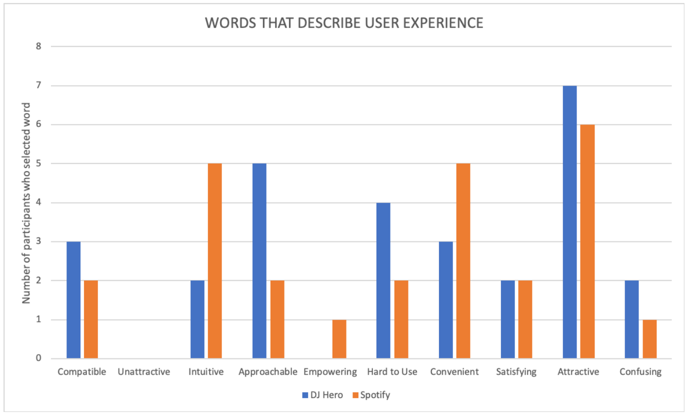


Figure 15: Users were asked to select three words from a given list that described the experience for each system.

Our study aimed to discover whether our system could streamline the process of sharing music preferences in a way that was fast and convenient. Regarding speed, results indicate that on average it takes longer to select preferences on our system than Spotify, however the mean times of the two tasks are within reach of each other according to the margin of error.

Regarding convenience, results indicate that while our system is attractive and visually approachable, the actual completion of tasks is confusing, whereas Spotify makes task completion more intuitive.

7 Discussion

Although the average time for Spotify task completion is much lower, we cannot definitively claim that Spotify is more efficient. Because of the low number of participants and the high variation among completion times, the standard deviations for both UI's average times were high. Thus, we cannot make a conclusion based on the comparison of average times.

Moreover, there is a bias when making a direct comparison between Spotify and DJ Hero. The study assumed that participants already had a Spotify account, and users who already have Spotify accounts are more likely to be familiar with the interface and how to achieve goals without the help of any documentation. This makes the comparison to our system, which participants saw for the first time during this study, less meaningful. This bias is reflected in the average completion times and standard deviations, which are noticeably lower for Spotify.

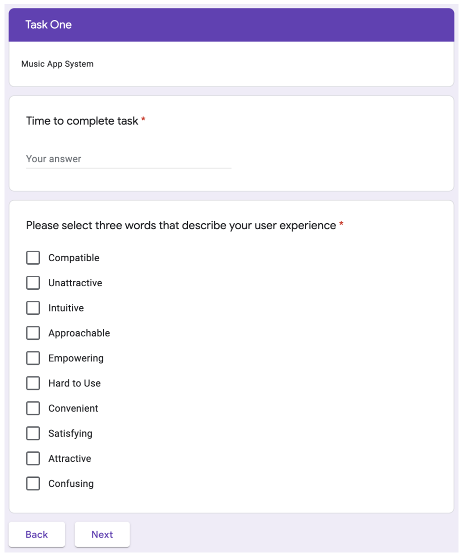
It is also not fair to label Spotify as a direct competitor. Although we used it for this study because it has some similar functionality, a future iteration of our system could see integration with Spotify. Even though for the purpose of this study we focused on curating music, a main functionality of it is that users can share preferences in a more streamlined and customizable fashion.

The system descriptors selected by users indicate that DJ Hero users have difficulty bridging the gulf of execution, and reveal problems with planning and completing actions. Thus, further work on DJ Hero is required in order to make tasks less confusing and more intuitive. Better signifiers are needed in order to make the affordances of the UI obvious and accessible to users.

After this study, we added more functionality and help documentation to our system. A newly added Help tab on each page gives users a place to go if they don't know how to achieve a goal. We also added more icons that indicate the state of the playlists getting edited, adding a check mark for saved items to give users a better sense of security.

Further implementation work would see expanded music and user databases. This could in part be aided by an integration with Spotify, using their large music and user databases with our tool that streamlines the social aspects of preference sharing. Further studies would examine the usability of the "Throw Party" function to see if potential party hosts find the functionality for obtaining friends' preferences easy and useful. We could also study the helpfulness of the "Discover Music" tab and could examine whether it influences the music tastes of individuals in a study group overtime.

A Appendix


Task One

Music App System

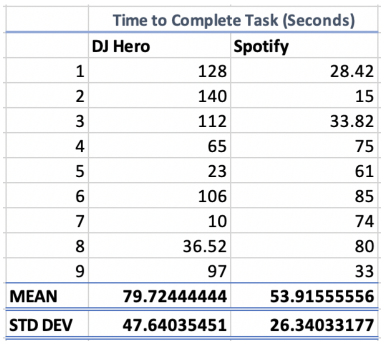
Time to complete task *

Your answer

Please select three words that describe your user experience *

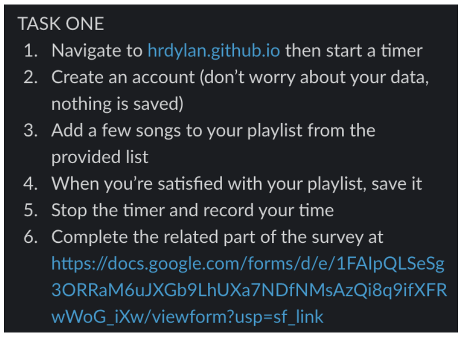
- [ ] Compatible
- [ ] Unattractive
- [ ] Intuitive
- [ ] Approachable
- [ ] Empowering
- [ ] Hard to Use
- [ ] Convenient
- [ ] Satisfying
- [ ] Attractive
- [ ] Confusing

Figure A.1: An example of the survey for a task. Each task requested the participant for the time it took to complete the task and 3 descriptor words, the order of which was shuffled for each task.



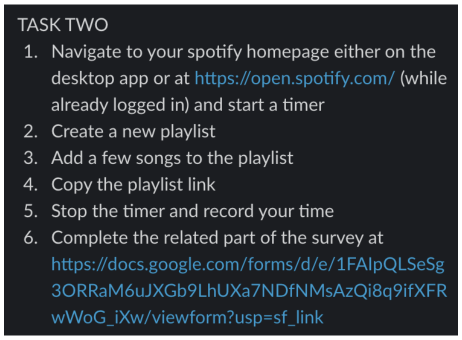
	Time to Complete Task (Seconds)	
	DJ Hero	**Spotify**
1	128	28.42
2	140	15
3	112	33.82
4	65	75
5	23	61
6	106	85
7	10	74
8	36.52	80
9	97	33
MEAN	**79.72444444**	**53.91555556**
STD DEV	**47.64035451**	**26.34033177**

Figure A.2: The full results for amount of time it took to complete the tasks.


TASK ONE

1. Navigate to hrdylan.github.io then start a timer
2. Create an account (don't worry about your data, nothing is saved)
3. Add a few songs to your playlist from the provided list
4. When you're satisfied with your playlist, save it
5. Stop the timer and record your time
6. Complete the related part of the survey at https://docs.google.com/forms/d/e/1FAIpQLSeSg3ORRaM6uJXGb9LhUXa7NDfNMsAzQi8q9ifXFRwWoG_iXw/viewform?usp=sf_link

Figure A.3: Instructions given for the DJ Hero task.


TASK TWO

1. Navigate to your spotify homepage either on the desktop app or at https://open.spotify.com/ (while already logged in) and start a timer
2. Create a new playlist
3. Add a few songs to the playlist
4. Copy the playlist link
5. Stop the timer and record your time
6. Complete the related part of the survey at https://docs.google.com/forms/d/e/1FAIpQLSeSg3ORRaM6uJXGb9LhUXa7NDfNMsAzQi8q9ifXFRwWoG_iXw/viewform?usp=sf_link

Figure A.4: Instructions given for the Spotify task.

B Acknowledgements

We'd like to thank the members of CSCI 3715 for participating in our study and for providing feedback along the way. Also thank you Bootstrap and Confetti-Canvas.

References

[1] Moore, Connor. "You'Re Not That Woke: on the Importance of Music, Culture and Self-Doubt." The Orient, 26 Apr. 2017, bowdoinorient.com/2017/04/26/youre-not-that-woke-on-the-importance-of-music-culture-and-self-doubt/.

[2] J. Aucouturier and F. Pachet, "Scaling up music playlist generation," Proceedings.IEEE International Conference on Multimedia and Expo, Lausanne, Switzerland, 2002, pp.105-108 vol.1.