DalSegno: User-centric preference elicitation strategies for mitigating cold start in music recommender systems

by

Cynthia Lin

S.B. in Electrical Engineering and Computer Science
Massachusetts Institute of Technology (2023)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2024

© 2024 Cynthia Lin. All rights reserved.

The author hereby grants to MIT a nonexclusive, worldwide, irrevocable, royalty-free license to exercise any and all rights under copyright, including to reproduce, preserve, distribute and publicly display copies of the thesis, or release the thesis under an open-access license.

Authored by: Cynthia Lin
Department of Electrical Engineering and Computer Science
January 26, 2024

Certified by: Eran Egozy
Professor of the Practice, Music and Theater Arts
Thesis Supervisor

Accepted by: Katrina LaCurts
Chair, Master of Engineering Thesis Committee

DalSegno: User-centric preference elicitation strategies for mitigating cold start in music recommender systems

by

Cynthia Lin

Submitted to the Department of Electrical Engineering and Computer Science
on January 26, 2024, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Avid music enthusiasts often rely on music recommender systems to sift through expansive music catalogs and find new songs fitting their interests. However, such systems struggle to personalize suggestions for new users as they heavily rely on extensive listening histories to make accurate suggestions — an issue known as the new user cold start problem. This problem is exacerbated by the fact that most commercial recommender systems lack transparency and avenues for users to influence their recommendations.

We thus propose **DalSegno**, a music recommender system with an interactive web-based user interface. The platform is designed to overcome the new user cold start problem by iteratively presenting users with recommendations and incorporating elicited feedback. Additionally, **DalSegno** enables users to learn about and fine-tune their inferred music preferences through interactive visualizations of song characteristics.

Throughout three rounds of user testing, **DalSegno** demonstrated promising results. Participants appreciated the system's ability to incorporate user feedback to provide more relevant recommendations and considered it more intuitive to use than commercial recommendation systems. Additionally, users felt that the interactive visualizations of musical qualities helped them learn more about their personal music tastes, which encouraged them to further utilize the interface. Overall, positive evaluations of **DalSegno** demonstrate that incorporating user input and fostering explainability is vital to creating a more user-focused and effective music discovery experience.

Thesis Supervisor: Eran Egozy
Title: Professor of the Practice, Music and Theater Arts

Acknowledgments

This thesis represents the culmination of my experiences at MIT, but it would be impossible to express gratitude to everyone I've met in the past 4.5 years here. However, this work surely would not exist without the unwavering support of my advisor Eran, who guided me in nurturing a mere figment of my imagination into a full-fledged thesis project. I serendipitously first met Eran through a friend's recommendation to enroll in 21M.387 in the fall of 2021. I knew that there were a lot of cool and smart people that I could take classes from at MIT, but I couldn't have fathomed that I'd be begging for pset extensions from the creator of one of my favorite childhood party games: Guitar Hero! Eran has been everything that I'd want in a mentor — candid yet approachable, gentle but firm, imaginative yet pragmatic, and both curious and wise. Given that my MEng journey has been everything but a walk in the park, there were moments when I doubted if I'd be able to finish, but Eran's faith in my abilities has never faltered. I consider myself fortunate to crossed paths with someone who shares my penchants for music, video games, good food, and dry humor. Eran's accomplishments will always inspire me to innovate and motivate others as profoundly as he has.

I'd also like to thank Eran for introducing me to Zahra Nazari of Spotify; I learned a lot about Spotify's recommendation systems from Zahra in our brief meeting together, and she shared valuable insights that were critical for paving the underlying groundwork for this project.

Thank you to the members of the MIT Technology Lab — Julie, Ajay, Matt, and Rachel — for not only sharing new perspectives and potential solutions during group meetings, but also for participating in the alpha tests. I'd also like to similarly thank all of the users who tested the beta and final prototypes.

To many Travelers and Warriors of Light: Thank you for assisting my free-to-play accounts in co-op content. I'd like to especially thank members of MIT «IHTFP» for powerleveling me through duty roulettes and Eureka.

To my fellow discord.js staff and developer community members: thanks for keeping me posted on updates in the Node.js and Discord ecosystems over the past 7 years, alerting me whenever a new hot and shiny framework has been launched, and encouraging me to improve my development skills.

To Mel, Fernando, Alicia, Margaret, Joanna, Grace, Naomi, Lucas, Kiet, Dianna, Will, Amish, Lena, Agustin, Cecilia, Cameron, and many more friends: thanks for hanging out with me throughout these past 1.5 years and providing me with fun but brief respites from the harsh reality of an impending thesis deadline.

To Jia Xin: jxbbxxnpzwzjnyybzwyyglwjxbbwan!

To my younger sister, who shares my love for cursed memes and vitriol for loud noise flops and fat squirrels: no u (there's so much more in life that you'll be exploring now that you're in college, so I'm excited to see where your journey takes you).

Finally, to my parents: thanks for always supporting my ambitions in music and technology. Even if we disagree at times, I understand that you want what's best for me, and I appreciate all your hard work in helping me get to where I am today.

Contents

Chapter 1

Introduction

Recommender systems are designed to filter through large sets of data, identify items that best align with user preferences, and present the most interesting content to users. Specifically, music recommender systems deployed on streaming platforms like Spotify and Pandora can tailor song and artist recommendations to introduce users to new music they might enjoy listening to.

Recommender systems generally have several established approaches to filter data. Content-based filtering involves recommending items with similar features to those a user has previously expressed favorable interest in. Music recommender systems following this approach may take into account a combination of low-level audio features and metadata to infer high-level descriptors capturing a user's music preferences [3]. On the other hand, collaborative-based filtering systems take the collective preferences and feedback of like-minded users into account when making recommendations to a target user. Such aggregate data includes crowd-sourced song descriptors or user listening histories and heavily contributes to the effectiveness of these systems' recommendations.

For new users, however, recommender systems struggle with making personalized recommendations. Compared to extensive preferences collected from long-time users who have spent hundreds of hours on a platform, a recommender system knows little about what a user likes or dislikes when they first enter the system, even if it prompts the user to express their preferences when they first register on the platform. As a result, the recommender system often makes false inferences from the limited data on the new users' preferences, and thus inaccurate recommendations, which leads to users having a poor experience. This phenomenon faced by recommender systems is known as the *new user cold start problem*.

We propose **DalSegno**, a music recommender system sourcing data from Spotify that utilizes a novel hybrid content-based and collaborative-filtering approach to suggest songs to a user. The system presents its suggestions in a web-based user interface that provides additional context to new users by informing them what qualities each particular recommended song has. To alleviate the cold start problem for the recommender system, the user interface allows the user to weigh specific song attributes in the filtering algorithm more heavily than others, and query for new songs better matching that configured profile, so that the recommender system has a more explicit understanding of the user's preferences. Additionally, as users encounter songs they enjoy, **DalSegno** determines which songs have similar characteristics together so that users can understand trends in their preferences and easily export them into Spotify playlists.

Chapter 2

Related Work

The problems of learning new user preferences, making better-informed recommendations accordingly, and visualizing them in interactive interfaces have been explored in a variety of contexts. The following is a brief overview of research done in the area of music recommender systems.

2.1 Early work in music recommender systems

Much of the early research surrounding music recommender systems started in the early 1990s when the Internet became an established avenue to digitally store, distribute, and discover music. Instead of proactively recommending music to users, these early systems allowed users to search and filter music by their factual metadata, such as their bibliographic information (e.g. song titles, artists, and albums) [3]. These systems worked well on smaller music catalogs but failed when metadata could not be properly standardized as music catalogs grew larger. Additionally, the early systems lacked the capabilities to introduce users to new music that they didn't know about.

Ringo (1995) was one of the first systems to directly address this issue by taking a "social information filtering" approach to make personalized music recommendations [10]. Ringo suggested artists liked by other users with statistically similar tastes in music, which inspired methodologies for future collaborative filtering systems. However, new users struggled with Ringo because they did not receive accurate recommendations until after rating hundreds of artists on the platform, exposing the system's weakness to the cold start problem. This suggests that Ringo's onboarding process, which consists of asking users to rate a semi-randomized collection of artists, could be improved.

2.2 Content-based filtering techniques in music recommender systems

Modern content-based music recommender systems augment factual metadata with analyses of low-level audio features to gain additional insight into a user's music preferences. For instance, in 2002, Kuo and Shan developed an algorithm that analyzed pitches in a song to determine its chord progressions and melodic patterns, which was supported by a recommender system that could suggest tracks fitting a certain genre based on their melodies [5].

Another content-based recommender system developed by Bogdanov et al. in 2013 computed semantic descriptors (i.e. genre, mood, rhythm) for a sampling of tracks in a user's music library based on various timbral, temporal, and tonal features, such as pitch, beat onsets, and spectral complexity [1]. These descriptors were augmented with bibliographic metadata sourced from the MusicBrainz Picard database and mapped into a similarity space. Descriptors were calculated similarly for each track in a music collection, and the system experimented with different similarity measures to suggest tracks in the music collection that were closest to the user's sample tracks in the similarity space. The system demonstrated its resilience to the cold start problem by its ability to infer user preferences and make successful recommendations for tracks the user had not encountered before. However, users generally preferred recommendations made by a different commercial system utilizing black box collaborative-filtering techniques.

In 2021, Okada, Tan, and Tamioka proposed a similarly content-based system that could map songs to a spectrum of user preferences based on their audio timbral features; however, their system differed in classifying tracks according to the MUSIC model, a psychology-based framework characterized by the five attributes of Mellow, Unpretentious, Sophisticated, Intense, and Contemporary [7]. Additionally, the system took into account users' psychological profiles such as age, empathy level, and tendency to think systematically, which were inferred from their responses to a self-assessment questionnaire, when making recommendations to them. Experimental results demonstrated that both age and personality were good initial predictors of a user's inclination towards mellow, unpretentious, or intense music and thus make appropriate recommendations, showing that the system was resilient against cold start. However, the system struggled with recommending sophisticated and contemporary music, hinting that additional audio features may need to be taken into account to properly determine the higher-level, culturally-defined properties of these tracks. Also, the validity of the empathizing-systemizing theory inspiring the system's user psychological profiling methodology has been widely questioned due to its inability to be replicated and its perpetuation of harmful gender stereotypes [11], leading us to wonder if better results could have been achieved by using a better-established personality-profiling framework.

2.3 Collaborative-based filtering techniques in music recommender systems

As demonstrated in the previous section, content-based recommender systems have great potential for capturing the musical qualities of a song, but may struggle to recognize higher-level qualities subjectively perceived by humans. Collaborative-based recommender systems rectify these shortcomings by analyzing patterns across a variety of user habits and behaviors, which allows them to make more nuanced inferences about how a song aligns with a user's complex musical tastes that differ across their entire music library.

One popular social media platform aggregating music listening habits is Last.fm, which tracks user listening histories to inform users and their friends about their unique listening patterns. In 2012, Purushotham, Liu, and Kuo trained a Bayesian model to interpret a Latent Dirichlet allocation (LDA) statistical model of user track ratings and a matrix factorization of their Last.fm data to provide music recommendations [9]. They discovered that their system made the most accurate recommendations when both user ratings and social network data were taken into account, noting that it was more important to consider personal ratings over friends' preferences. While lacking significant user rating data, the system made more accurate recommendations when friends' listening habits were taken into account because users tend to appreciate similar music as their friends. However, it made a greater number of successful recommendations when more user rating data was provided.

Streaming platforms such as Deezer also maintain similar logs of their users' listening histories as Last.fm. In particular, Briand et al. designed a system that clustered long-time Deezer users into groups based on their demographics (age, country) and various listening habits, such as streaming activity, searches, skips, and likes in 2021 [2]. When a new user registered on Deezer, they were matched to the cluster of users who had the closest centroid to the predicted embedding vector of the new user's self-reported demographics and day-one usage data. This semi-personalized strategy performed better than the ones that recommended the most popular songs to new users or songs that matched strictly the new user's listening preferences, and deploying this recommender system onto Deezer resulted in a higher number of streams and liked songs across the platform.

Both collaborative-based recommender systems demonstrate that although user rating data is important, additional nuances about their listening preferences can be inferred by comparing them to other longtime, like-minded users. However, since these collaborative-filtering systems need a significant amount of data about many users to succeed, it would take considerable time and resources to develop a system that could replicate similar results.

2.4 Other considerations for music recommender system design

Some other music recommender systems consider additional factors when giving suggestions to users, especially to mitigate cold start problems. For instance, a system designed to make recommendations for tracks to be added to a playlist extracted playlist titles and analyzed them for additional meaning [13]. Another system, recognizing that users might prefer to listen to different types of music as they performed various activities throughout the day (e.g. studying, working out), took into account sensor data from mobile phones to predict what a user was doing and made more targeted recommendations [12]. Finally, a recommender system assessed whether users were receptive to divergent recommendations by measuring if they altered or decreased their activity on the platform upon encountering songs that didn't align with their expressed preferences [6]. This system succeeded in increasing user engagement on the music streaming platform Spotify by making more divergent recommendations to open-minded users while keeping recommendations consistent with expressed preferences for less-receptive users.

2.5 User interfaces supporting music discovery

User interfaces for music collections often employ unique visualization techniques to represent the qualities of each track in a given collection [3]. Knees et al. discovered several interfaces that map songs onto a multi-dimensional space so that similar tracks are grouped closer together [4]. For example, they encountered an early interface developed in 2001 that colored songs according to their genre classifications, and then projected them into a 3D space based on the similarity between their audio feature vectors. Another set of interfaces from the 2000s took a geographically-inspired approach of representing clusters of self-similar songs as islands or mountains on a map; one particular interface from 2009 even allowed users to edit the map to correct the recommender system's perceived similarity. Knees et al. also found other interfaces that map tracks, artists, or genres onto wheel-shaped, spherical, and galaxy-inspired spaces.

Some interfaces are particularly designed to encourage divergent exploration of a music collection, which is especially promising for solving cold start problems as they can help recommender systems collect new data about user preferences. One such interface in 2005 allowed users to interact with a flow of discs — each representing a track — to discover tracks similar to ones that they had actively selected [3]. To support a collaborative-filtering system and collect metadata for songs, another user interface made in 2007 challenged users to quickly and correctly identify a given song by adding appropriate semantic tags [4].

Others have also investigated unique ways to represent user music preferences so that users can not only better understand them, but also gain additional context from them. In particular, the 2013 recommender system developed by Bogdanov et al. was supported by a visualization of user preferences as humanoid cartoons, such that each body part or clothing item on the avatar captured some aspect of their music preferences (e.g. genre, danceability, instrumentation) [1]. Although it is unique and humorous, such a generalized and metaphorical visualization might not completely capture all the nuances of a user's music preferences, and it is not immediately obvious what certain features (especially body parts) are supposed to represent.

Most recently in 2022, Petridis et al. developed an especially promising interactive web interface visualizing user music tastes as a traversable network of artists, aptly named TastePaths [8]. Specifically, each artist is represented as a colored node in a graph according to the cluster of genres they belong to, and they are linked to similar or related artists through graph edges. Prior to generating a graph, the interface asked users to identify genres users listened to often and artists within that genre to serve as "anchor" nodes. Artists in the graph were clustered through the Louvain graph clustering algorithm, and common genres unique to that cluster were identified through term frequency-inverse document frequency (TF-IDF). Artists were linked through a shortest paths algorithm — specifically, a bidirectional version of Dijkstra's algorithm — from the most popular anchor artist to each of the other anchor artists, such that all the intermediary nodes from these shortest paths are added to the initial graph and each node has two edges. The interface was successful in helping users identify subgenres in their music preferences and discover new artists that combined different musical styles they liked. However, users wished that the graph conveyed additional context, such as which artists they had already listened to and historical evolutions of genres, and additional interactivity with the graph so that they could prune uninteresting artists or more deeply explore connections to a particular artist. These flaws suggest that a graph-based visualization may not necessarily be the most effective way of presenting the information collected by the recommender system.

Chapter 3

Design

DalSegno is designed to resolve the user cold start problem by employing a hybrid recommendation approach, improving system explainability and transparency, and utilizing explicit and implicit user feedback to better understand a new user's preferences. It interfaces with the music streaming platform Spotify to access established sources of user community and audio analysis data.

3.1 Design Goals

DalSegno encourages users to explore and discover new songs they'll enjoy in accordance with the following goals:

3.1.1 User autonomy

Users should be free to accomplish a variety of complex tasks that allow them to better discover or understand their music tastes. For example, they should be able to delve deeper into a track of interest by searching for songs with similar musical qualities with a slight difference in a particular attribute. Users should be able to affirm results from the recommender system by noting which types of songs the system should prioritize looking for, or reject suggestions and query for different ones. However, the interface needs to be designed such that the system will not misinterpret overly detailed or conflicting demands from the user. Regardless, we believe that providing these capabilities is important to making the system more resilient to cold start by operating off of data from active user engagement with the recommender system.

3.1.2 Ease of use

Users should not struggle to understand how to navigate or use the interface. Features should be discoverable and intuitive to use. The interface should be designed to allow users to easily execute tasks. We hypothesize that users will find the interface more engaging if they actively drive their own music discovery. The associated design challenge with this ideal manifests in how much guidance or assistance we need to provide the user to help them navigate the interface seamlessly. Whereas some users may prefer to self-explore, other users may prefer more guided, interactive experiences.

3.1.3 Comprehensibility

The user interface must allow users to understand their established music preferences; users unfamiliar with their own music tastes should be able to confidently describe the types of music they like after using the platform, while more musically attuned users should be able to discover minor nuances about their music that they did not previously realize.

One particular design challenge is how to handle the disparate music preferences a user might have. Users may listen to different types of music and thus create queries involving multiple genres. In order to find recommendations that fit a varied palate, we need to be able to distinguish disjoint subsets of songs that share similar musical traits. Doing so allows us to find songs that closely match each identified preference and equally represent all of a user's music interests in the resulting recommendations, thus improving personalization.

3.1.4 Robustness against user cold start

Although our system will have knowledge of which tracks or artists a user has often listened to from Spotify, such data may not necessarily indicate or encompass a user's listening preferences if they rarely listen to music on Spotify. Thus, we hope to investigate which of the various techniques we employ in this project are effective in making more relevant recommendations for new or cold users. Specifically, we explore the effectiveness of a strategy of directly eliciting preferences from users by enabling them to reweigh parameters in the recommendation algorithm to better fit their priorities.

3.2 Data and Parameters

It takes a significant amount of time to start a music collection from scratch, compute multiple measures of complex musical qualities on each track from raw audio data, and track user listening histories across the music collection. In other words, we face *new item and community cold start problems* attempting to generate our own audio analysis and user community data, which is out of the scope of this project. Thus, we heavily rely on various data provided by Spotify's proprietary systems.

3.2.1 Audio analysis data

Audio analysis parameters taken into account by the recommender system are sourced from Spotify's audio features endpoint[1], which calculates audio features for each track in their music collection. The musical qualities obtained in this manner and utilized in our recommendation algorithm are:

- *acousticness* — a measure of how naturally sounding (as opposed to being produced by an electronic or synthesized instrument) a piece of music is

[1] https://developer.spotify.com/documentation/web-api/reference/get-several-audio-features

- *danceability* — a measure of how suitable a song is for dancing based on perceived attributes related to rhythm
- *energy* — a measure of the levels of intensity and activity in a song
- *tempo* — the speed or pace of a given piece and derives directly from the average beat duration, typically measured in beats per minute (BPM)
- *valence* — a spectrum of musical positiveness conveyed by a track, which can range from happiness to sadness

These parameters are also utilized in Spotify's track recommendation endpoint[2] that returns tracks matching the specified audio analysis parameters. We use this endpoint to query for tracks with similar qualities to a target track the user has identified by specifying the `target` threshold for each quality. If the user queries with multiple target tracks, we identify trends within the inputs and query with those self-similar subgroups instead.

Although other features, such as loudness, liveness, and speechiness, were exposed through the audio features endpoint, they were not taken into account by the system as they had negligible impact on the effectiveness of recommendations and visualizations.

One potential tradeoff of using this data source for audio analyses is that Spotify does not provide precise details on how each quality is computed, which makes it difficult to evaluate the accuracy of the derived measures. We plan on countering this by augmenting the interface with explanations and disclaimers of what we know about how each measure is calculated.

3.2.2 Audio metadata

Bibliographic catalog information, or metadata, for a track, such as release year, album cover art, and performing artists, is available at Spotify's general track API

[2] https://developer.spotify.com/documentation/web-api/reference/get-recommendations

endpoint[3]. The information for each track can be accessed by providing the unique base-62 Spotify ID[4] associated with the track. To retrieve Spotify IDs based on metadata (such as when a user provides a track name to use as a seed), we can access the track payloads delivered by endpoints such as the generic item search endpoint[5]. We display such metadata on the interface to help users uniquely identify and recognize tracks.

3.3 User experience

The **DalSegno** experience is an iterative process consisting of a few key parts: user onboarding, initial seed selection, recommendation browsing, preference summaries, history review, and finally playlist export.

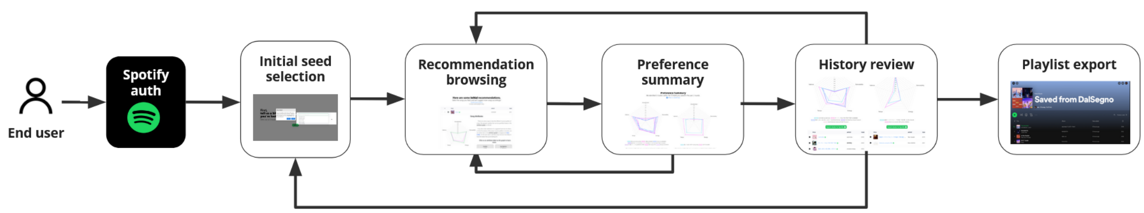


Figure 3-1: Overview of the **DalSegno** user experience.

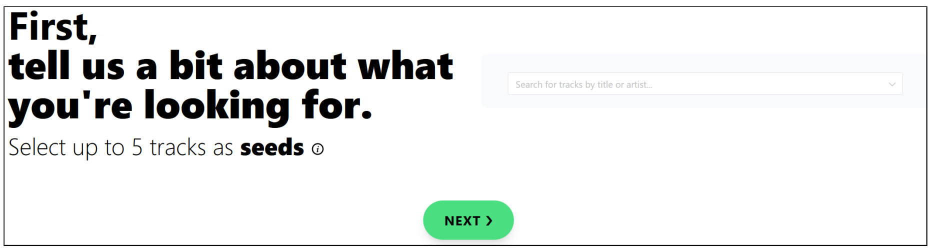
3.3.1 User onboarding

Starting from the landing page, new users begin their experience by authenticating with their Spotify credentials. Authorizing **DalSegno** to access their Spotify account data enables a more personalized and integrated recommendation experience. Afterwards, users are redirected back to the landing page, which is updated to show the user's basic profile info to affirm that they have successfully onboarded.

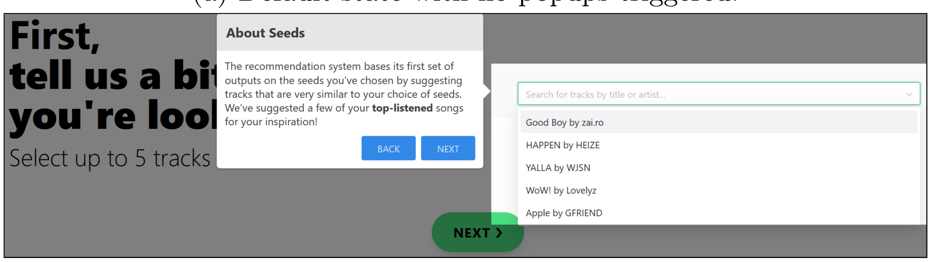
[3]https://developer.spotify.com/documentation/web-api/reference/get-track
[4]https://developer.spotify.com/documentation/web-api/concepts/spotify-uris-ids
[5]https://developer.spotify.com/documentation/web-api/reference/search

3.3.2 Initial seed selection

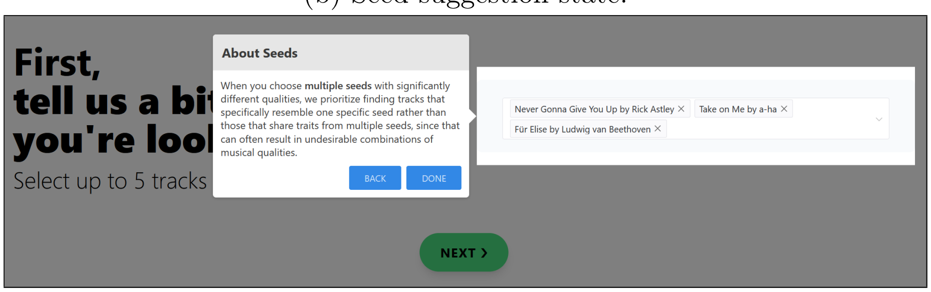
Next, users are prompted to select initial seeds that serve as the starting point for the music recommendations in their experience (Figure 3-2a). A series of interactive popups introduces the concept of seeds and provides further context by explaining how they are used by the algorithm to generate personalized recommendations. Users are suggested potential seeds based on their listening history (Figure 3-2b), but they can select other seeds by querying based on track or artist name (Figure 3-2c).



(a) Default state with no popups triggered.



(b) Seed suggestion state.



(c) Multiple-seed selection state.

Figure 3-2: Initial seed selection interface in various states.

3.3.3 Recommendation exploration

Users are presented with a set of recommendations based on the selected initial seeds (Figure 3-3). They have the option to explore these recommendations by listening to

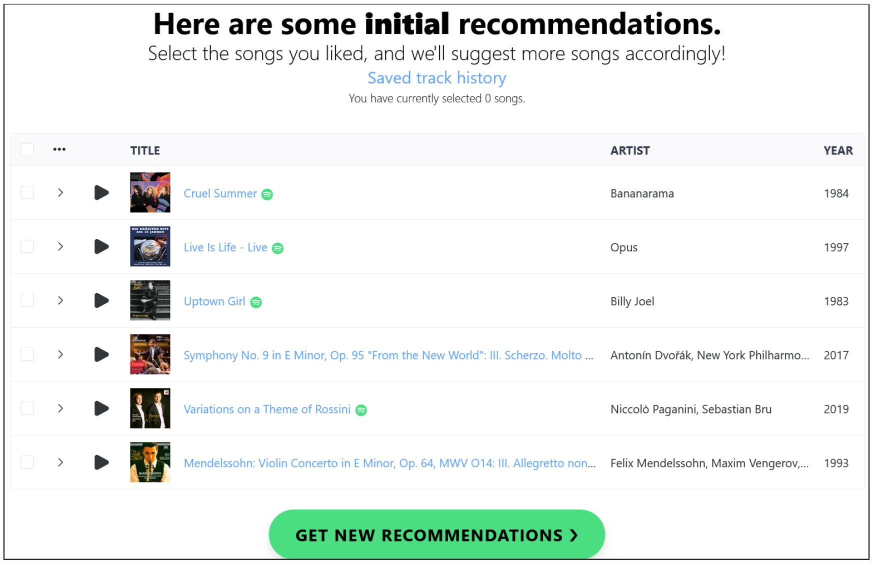
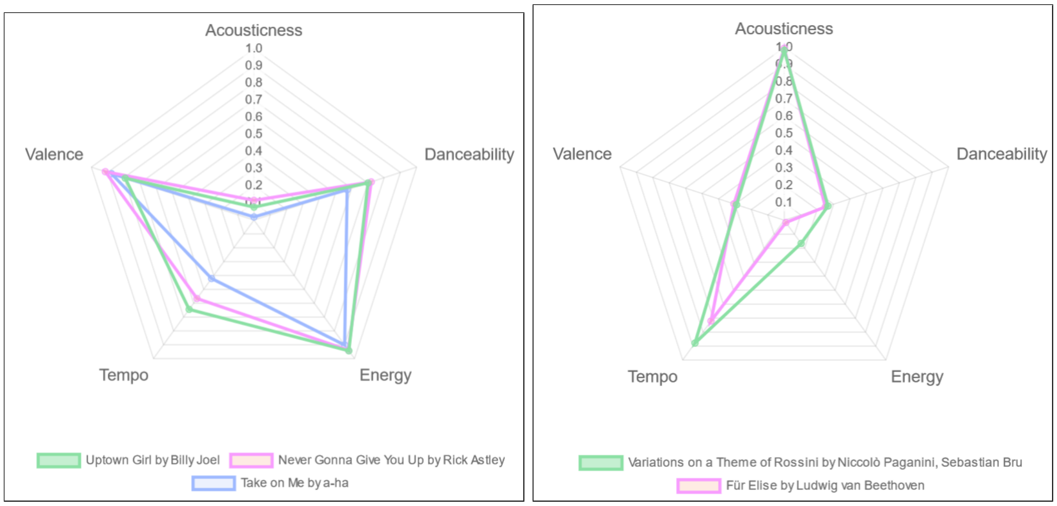


Figure 3-3: Sample set of recommendations.

30-second audio previews by clicking on the play icons, and clicking on the song title user's native Spotify client.

To provide positive feedback to the recommendation algorithm, users can *save* tracks to use as seeds for the next set of recommendations by selecting their corresponding checkboxes. If users do not encounter recommendations that they like, however, they can generate a new set of recommendations for the same seeds by clicking the GET NEW RECOMMENDATIONS button without choosing any tracks as seeds.

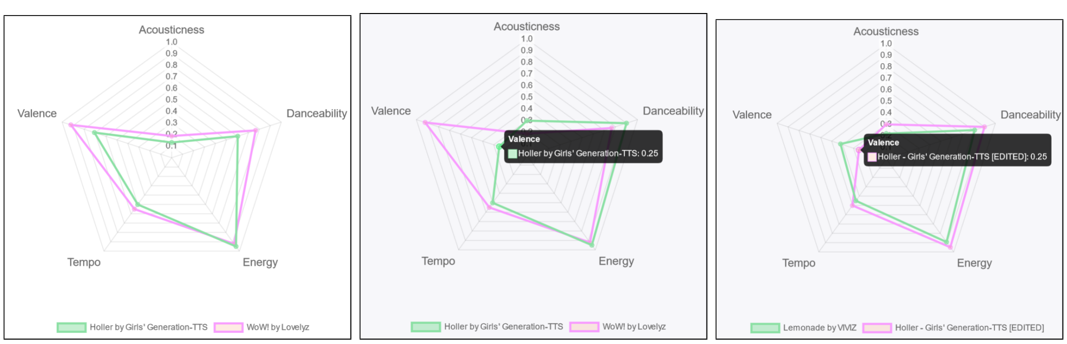
Users can gain a more comprehensive understanding of their music preferences by exploring *attribute graphs* that accompany each recommendation. These visualizations depict various musical qualities — acousticness, danceability, energy, tempo, and valence — of each recommendation. They can also learn more about how each quality is defined by clicking its label on the graph. Furthermore, **DalSegno** helps users understand their diverging preferences by visualizing attributes relative to those of similar seeds. For example, in Figure 3-4a, the recommendation "Uptown Girl"'s attributes are displayed relative to similar seeds "Never Gonna Give You Up" and "Take On Me". However, the attribute graph of "Variations on a Theme by Rossini" indicates that it has a distinct profile that more closely resembles the seed "Fur Elise" than the other seeds (Figure 3-4b).



(a) "Uptown Girl". (b) "Variations of a Theme of Rossini".

Figure 3-4: Attribute graphs for two recommendations with different characteristics. Each colored line on the graph represents the characteristics of a recommendation or seed.

Attribute graphs also enable users to fine-tune their inferred preferences. By rearranging the positions of nodes on an attribute graph, users can influence their subsequent set of recommendations to better match a different combination of musical qualities (Figure 3-5).



(a) Original attribute graph for "Holler". (b) "Holler" attributes after user modification. (c) Recommendation based on modified "Holler".

Figure 3-5: Demonstration of the attribute editing process on attribute graphs.

3.3.4 Preference summaries

After a few rounds of recommendations, **DalSegno** provides a summary of the preferences it has elicited from the user (Figure 3-6). Like the recommendation browsing phase, preferences are visualized through attribute graphs, and brief textual descriptions are also generated. Users can proceed to review their seed history, or continue refining their preferences through additional rounds of recommendation browsing.

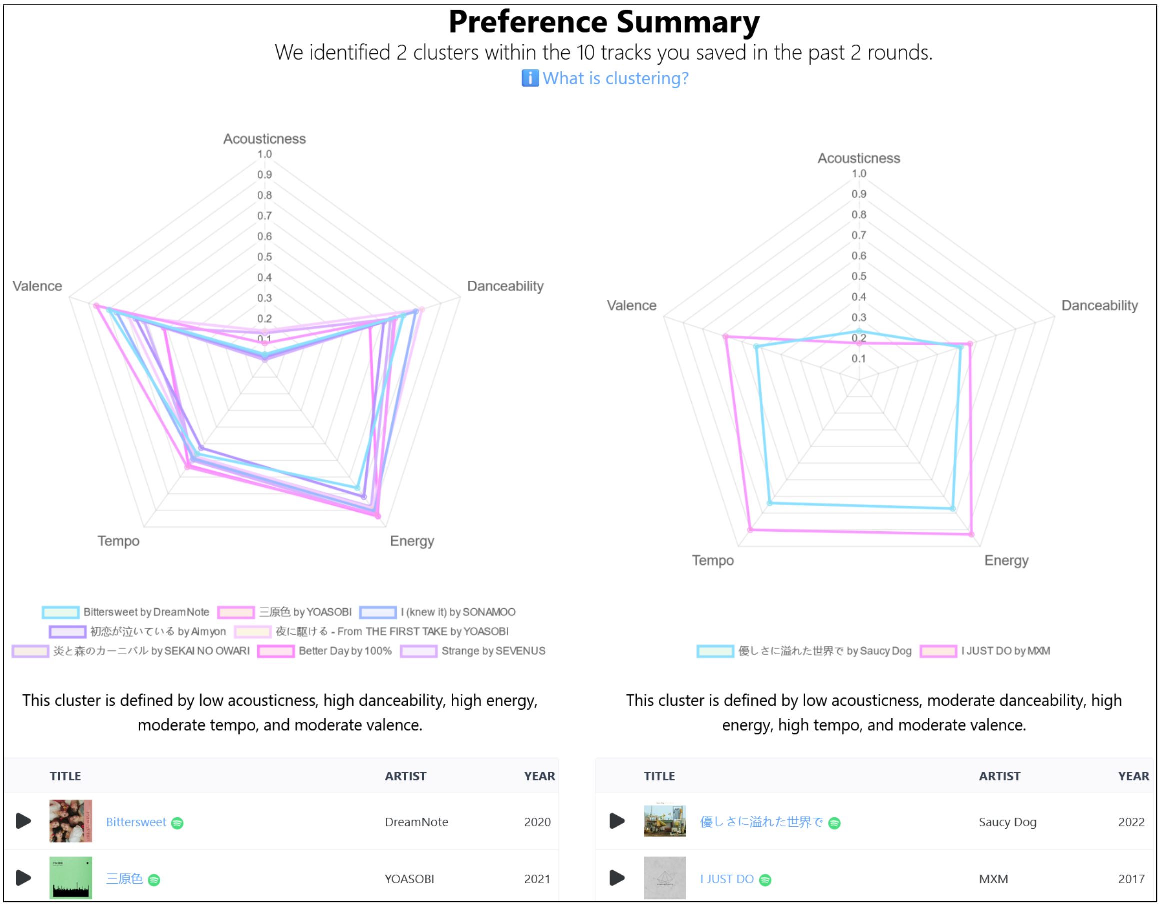
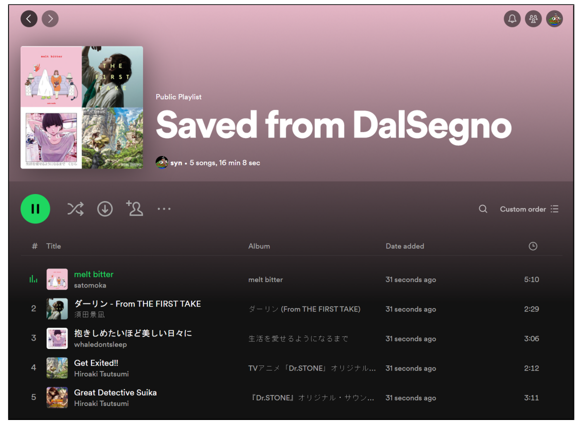


Figure 3-6: Example of a preference summary generated after the user completed two rounds of recommendations.

3.3.5 History review

As the final phase of the **DalSegno** experience, users can review all the seeds they saved thus far. Similar seeds are grouped together, while seeds with distinct characteristics form their own separate clusters (Figure 3-7). Additionally, users can export each group of songs into a Spotify playlist that they can listen to later (Figure 3-7b). They can finish their experience by either initiating a new one with a different set of initial seeds or logging out, which results in all of their session data being deleted.

(a) Two distinct clusters of songs.



(b) Exported playlist for the left cluster in Figure 3-7a.

Figure 3-7: History page overview for 7 saved seeds.

Chapter 4

Implementation

The **DalSegno** platform is implemented as a single-page web application. Its modules utilize various data analysis packages and web frameworks from the Python and Node.js ecosystems, respectively. Ultimately, the components integrate seamlessly to enable an interactive user experience that allows users to easily see, hear, and discover new music. The source code for **DalSegno** is accessible through this GitHub repository:

https://github.com/synicalsyntax/dalsegno

The system was designed to be self-hosted on a virtual private server. A public instance of **DalSegno** is deployed here:

https://dalsegno.syncl.dev

4.1 System architecture

DalSegno consists of several components, including a Vue.js frontend, Flask backend, and various data processing modules written in Python. The system is designed so that the Flask backend facilitates communication between the frontend and backend through a REST API, while the user interacts with the system primarily through visual elements afforded by the frontend. Each module on the backend serves a different purpose: for example, one module is dedicated to handling the OAuth authorization flow described in Section 4.3.1. A diagram of the system architecture is pictured in Figure 4-1.

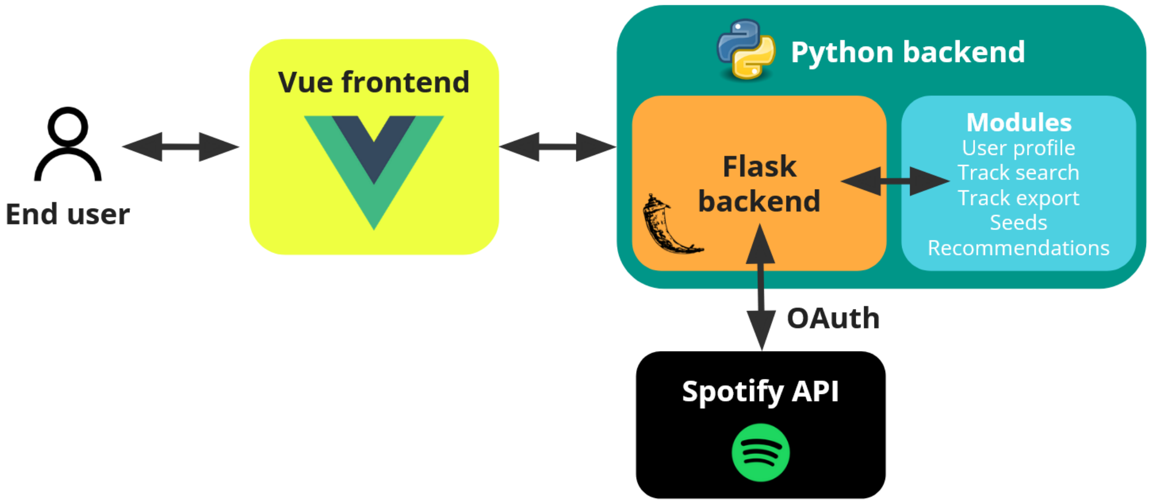


Figure 4-1: Overview of the system components and data flow

4.2 Vue frontend

Vue.js (also referred to as Vue) is a framework designed for building user interfaces. Although the long-running 2nd version of Vue arguably has a larger package ecosystem, we opted to use the 3rd version of Vue as it offers first-class support for TypeScript. The code for the frontend is available in the `src` folder.

4.2.1 Modular components

Vue enables a component-based architecture, such that each element can have its template markup, logic, and functionality encapsulated in a dedicated component and reused across different pages. These components are available in the `src/components` folder. Several frameworks streamline the development of components and are listed in Appendix A-2.

4.2.2 Client state

Contrary to most client-server implementations, **DalSegno** runs a stateless backend; instead of using server-side storage to maintain state, all session data is stored in the end user's browser, which thus allows them to maintain control over their data retention. This is accomplished through the state management library `pinia`, which closely follows Vue reactivity and composition paradigms. As a result, calls from the Vue frontend to the backend often encode several components of the user's client state.

4.3 Flask backend

The Flask backend implements a REST API that is responsible for delegating communication between the Vue frontend and Python modules on the server. Files related to the Flask backend are placed in the root of the `server` directory. Further information regarding the Python libraries used in this project can be found in Appendix A-1.

The backend closely integrates with the `spotipy` API library for Spotify to utilize convenient methods for common Spotify operations, such as fetching basic user profile data, searching for tracks, and fetching recommendations. Notably, `spotipy` also provides methods implementing Spotify's OAuth protocol with the PKCE variant, allowing **DalSegno**'s API client to make calls without its client secret; this ensures the security of our platform as sensitive credentials are not passed in any API calls.

4.3.1 Data access

The Spotify API implements the OAuth authorization protocol[1], in which users consent to Spotify-interfacing applications like **DalSegno** accessing specific types of data by granting *scopes*. As part of its user onboarding process, **DalSegno** requests users to grant the following scopes:

- `user-read-email` — obtain the name and profile image associated with a user's

[1] https://developer.spotify.com/documentation/web-api/concepts/authorization

Spotify account to display on the interface after authentication

- `user-top-read` — access a user's top artists and tracks to use as suggestions for initial recommendation seeds

- `playlist-modify-private` — create and modify private playlists on the user's behalf to help users export their saved tracks

Once a user grants access to their data, Spotify will redirect the user back to **DalSegno** and deliver a temporary *access token* to a dedicated endpoint on the **DalSegno** web server so that it can make API calls on behalf of the user, whether it be displaying the user's basic profile information or searching for tracks of similar qualities. Ultimately, this process ensures that users can consent to **DalSegno** accessing only specific parts of their personal data, and also grants them the agency to revoke access permissions at any time through their Spotify account settings.

Although we access the authorized user's top tracks and artists, that data is not used when considering potential recommendations unless the user explicitly chooses one of their top tracks as an initial recommendation seed. Additionally, accessing a user's listening history requires a different authorization scope[2] that we do not request. Thus, we can consider **DalSegno** to be agnostic to a user's music preferences until they initiate the recommendations process.

4.3.2 API routes

API resources and their associated routes are located in `server/api.py` and are listed below. Notably, all of these routes require the end user to be authenticated to access them; to do so, the `spotipy` authentication manager checks its cache to see if there is a valid token associated with each request before it is delivered to the endpoints. Middleware redirects users appropriately if they attempt to access invalid routes or protected routes while unauthenticated.

- `/profile` — Endpoints for user profile data

[2] https://developer.spotify.com/documentation/web-api/reference/get-the-users-currently-playing-track

  - `GET /profile` — Returns the authenticated user's basic profile info (name, profile picture, ID)

- `/search` — Endpoints for searching Spotify
  - `GET /search/spotify` — Given a string query, return a set of tracks that match the query through track or artist name.

- `/seeds` — Endpoints for seeds
  - `GET /seeds/suggestions` — Suggest a set of initial seeds for the user based on their most-listened tracks.
  - `GET /seeds/clusters` — Given a set of seeds, return disjoint clusters of seeds considered to be similar to each other.

- `/recommendations` — Endpoints for recommendations
  - `POST /recommendations/results` — Given a set of seeds, return song recommendations that are similar to each disparate subset of self-similar seeds.

- `/export` — Endpoints for data export
  - `POST /export/cluster` — Given a cluster of seeds, export it into a new Spotify playlist under the user's account.

4.3.3 Authorization endpoints

On the other hand, authorization endpoints are located in the `server/auth.py` file. The following routes are implemented to align with the OAuth protocol:

- `/auth/login` — Redirects the user to a unique Spotify authentication page where they can provide login credentials and grant the requested authorization scopes.

- `/auth/logout` — Clears the user's session data so they can re-login to Spotify if desired.
- `/auth/spotify/callback` — Listens to requests from Spotify that provide an authorization code that allows the `spotipy` API client to make requests on behalf of the authenticated user.

4.3.4 Recommendation algorithm

We utilize the `scikit-learn` library's agglomerative clustering functionality[3] to group similar input seeds together based on the vectors formed by the five musical attribute values described in Section 3.2.1. Unlike other methods (like k-means clustering) that require particular parameters to be specified, agglomerative clustering is well-suited for our use case of sorting a variable number of samples into a variable number of clusters. A new cluster is formed if the minimum cosine distance between a certain attribute vector and all other attribute vectors in the cluster exceeds a set threshold. Clustering seeds in this manner allows us to identify a user's disparate music tastes and find recommendations that match each type of preferred music.

Once clusters of seeds are distinguished, we create *profiles* of each cluster by statistically analyzing the combinations of musical attributes present in the cluster. Through the `pandas` library, we calculate the minimum, maximum, and target values of each attribute, which is set to the 25th percentile, median, and 75th percentile of the numerical values, specifically. These profiles are used as parameters in a subsequent call to the Spotify recommendations endpoint to filter out data points with mismatching profiles in the response.

To further identify recommendations that best match the input seeds, we utilize the `annoy` library to find the nearest neighboring vectors of musical attributes to the input seed vectors based on angular distance. Data processing techniques were employed to specifically eliminate recommendations that were identical to the input seeds (but perhaps were duplicated on an album repackage), as well as avoiding

[3] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html

recommending tracks that were too different from the target profiles. The resulting recommendations are sorted by the number of seeds that match the profile of the recommendation, and then by the minimum distance difference from a seed vector to the recommendation vector. The final output from invoking the recommendations module contains data that can be visualized by `chart.js` in a custom radar graph, where certain values like tempo are normalized to be in a range of 0 to 1.

Chapter 5

Experiments

Three stages of user testing were conducted throughout **DalSegno**'s development. Features were tested with groups of increasing size, which allowed us to make iterative improvements to the system to better align with user needs and preferences.

5.1 Alpha Testing

The initial alpha tests were a series of *contextual inquiry* field studies conducted with a small group of users. Through these tests, we sought to better understand the thought processes of **DalSegno** users so that we could redesign our user interface to better align with real-world usage patterns. Additionally, we hoped to validate assumptions made during the design process; by seeing how users actually used **DalSegno**, we were able to either confirm or refute our initial design hypotheses. Ultimately, understanding the usage context of **DalSegno** led to the development of additional features designed to improve usability and encourage long-term usage.

5.1.1 Methodology

We recruited four participants from the MIT Music Technology Lab for our alpha tests. This meant that each participant had an extensive prior background in both technology and music theory, as all were EECS upperclassmen or graduate students that conducted research in music technology. Recruiting such a body of participants ensured that users could recognize if the system made false inferences about their preferences, and could also offer constructive feedback on how to best improve the user interface and system design.

As part of the recruitment process, participants were asked to complete a consent form that informed them of the nature of the study and what data would be collected (Figure B-1). Specifically, participants had to consent to having their session audio, screen footage, and interview responses recorded for reference in research publications.

A researcher individually met with a participant in a private office setting to conduct each session. Participants accessed an early prototype of **DalSegno** via a development server running on the researcher's personal laptop, which allowed the researcher to mitigate mid-session issues. At the start of the session, the researcher explained the purpose of the contextual inquiry and a broad overview of the platform, emphasizing that the focus was on observing their natural behavior instead of prompting them on their ability to perform specific tasks. Screen and audio recordings were initiated as the users began to navigate the interface; as they did so, we directly observed how users naturally interacted with **DalSegno**, taking detailed notes on their actions, behaviors, and any challenges they encountered. Participants were also encouraged to monologue and explain their thought processes as they navigated the interface. Afterwards, we conducted follow-up interviews to understand how the participants perceived their user experience and encouraged them to share feedback on and suggest improvements for the system.

5.1.2 Results

A high-level summary of each participant's session is provided below.

Participant A

This session lasted approximately 32 minutes. Participant A encountered no issues onboarding onto the **DalSegno** application. As they selected five initial seeds, the participant noted that based on the introduction on the seeds page, they were inclined to choose as many seeds as they could with diverging characteristics. They also shared that as a music producer, they chose seeds that belonged to "niche" genres so they could learn more about the soundscapes of those genres.

Around 40 recommendation results were generated in the first round based on the five seeds they chose. They commented that it wasn't clear what they were supposed to do upon initially seeing the interface. Additionally, Participant A did not find the song attributes panel of their own volition, and had to be guided by the supervising researcher towards it. However, they did find the feature "useful", and later noted that they considered Danceability the "most important" attribute for defining their music preferences.

Participant A clicked on the play button of the first track listed on the page, and listened attentively to the following song snippet, noting that it sounded "fine". They then proceeded to listen to several tracks in a similar manner, deeply engaging with and openly reacting to the songs they listened to. For example, they commented "This is fire" when they found a track they liked, but disapproved of another track by an artist, saying "Nah, they missed." Notably, they opened certain tracks on their personal Spotify app to listen to them in its entirety and add them to their playlists if they were particularly interesting, which revealed an opportunity to more closely integrate with Spotify.

Participant A shared that although they did not recognize the artist that they first listened to, their heuristic for choosing songs to listen to was based on recording artists; they seemed particularly excited whenever they encountered artists they recognized.

While choosing seeds for the second stage, we observed that Participant A took longer than other participants; they explained that they wanted to carefully influence their future suggestions with their seed choice, which demonstrated that they understood the concept of seeding. While completing the later stages, Participant A listened to the song previews more briefly than the previous round, and quickly made judgments about how much they liked their music. They later remarked that they had grown fatigued by the third round, likely caused by the large amounts of seeds they encountered.

Participant A estimated that they liked approximately half of the songs they engaged with. They noted that although they did not necessarily learn what music they liked, they did learn what they didn't like to listen to. They also commented that it was a bit difficult to tell how the suggestions had changed over the first and third rounds. They also suggested designing a more guided experience by having tracks directly presented to users instead of having to click on buttons to listen to their previews or view their visualizations. Despite their critiques, Participant A found that the system was "useful" and wanted to try out future iterations of it. Additionally, they later shared that one of the songs they had encountered through **DalSegno** ended up being one of their most-listened tracks of the year.

Participant B

This session lasted approximately 41 minutes. Similar to Participant A, Participant B stated their goal for **DalSegno** was to explore the genres of its recommendations. They chose five initial seeds that varied greatly from each other and stated their goal for future rounds was to drive it towards recommending more songs from the disco genre. While reviewing the resulting recommendations from the initial seeds, they felt that the recommendation system "didn't know what to do with me" because the resulting tracks had all sorts of characteristics that didn't resemble one individual seed or another. However, when they chose a single seed, they thought that the results were much more similar to the seed. This indicated that potential improvements to the recommendation algorithm would involve identifying distinct groups of similar seeds.

Amusingly, Participant B also wanted to "stress test" the interface and try to "break it" to expose unknown bugs. Their first attempt to do so was inserting Chinese characters into the text input on the initial seed selection page, but the interface handled it correctly. However, Participant B encountered a bug where identical songs were shown twice within the same round of results. Upon choosing a seed titled "10 hours of air conditioner music", they demonstrated that a song's duration could be displayed incorrectly by the interface.

Participant B offered several suggestions for making the interface more intuitive, like adding info tooltips next to keywords and labels to provide more context. As they did not discover the track visualizations section without external input, Participant B also recommended having small previews of the attribute graph in each track listing. They also wished there was a convenient way to initiate new seed selections from the recommendations page.

Regardless, Participant B thought the **DalSegno** experience was "fun" and "satisfying", as they liked engaging with the song attribute graphs and summaries: "Dragging [the graph nodes] is definitely the funnest part!"

Participant C

This session lasted around 35 minutes. Participant C ran into the recurring issue of retrieving too many recommendations when they submitted 5 seeds in the initial round and similarly felt that they were overwhelming. Regardless, in subsequent rounds, they "did vibe with the songs [they] got back", and felt that their suggestions sounded similar to the seeds despite being different tracks.

Like the other participants, Participant C did not naturally discover the attribute graph without guidance from the accompanying researcher. They also did not realize that clicking on the labels would reveal the definitions of the attributes either. They believed that adding tooltips would make the feature's interactivity more obvious. Participant C also questioned whether having attributes like BPM and loudness displayed with the attributes would be meaningful to users lacking musical background. Nevertheless, Participant C interacted with the attribute graph significantly more than others; they believed that "the attribute graphs characterized the songs well" and that they learned from them.

Participant C appreciated being able to retrieve new recommendations if previous ones were unsatisfactory. However, they thought the interface could provide additional utility for users by integrating more closely with the user's data. For example, they noted that they struggled to come up with good seeds during initial seed selection, and thus it would be ideal to have the interface provide suggestions for seeds.

Participant D

This session lasted 17 minutes. Participant D first chose 3 initial seeds. Upon reviewing the resulting recommendations, they noted that the suggested songs were from the same artists, likely because their seeds belonged to the same genre. They were highly satisfied with their suggestions, at one point exclaiming "Ooh, I like this!" Participant D also appreciated various other aspects of the interface, such as animations and color transitions, noting that the interface "looks like Spotify".

Participant D also struggled to find the attribute graph feature and didn't engage with it much after finding it. They also didn't notice that the panel would update with attribute definitions until it was explicitly pointed out to them. Like Participant C, they suggested changing the cursor shape around the labels would help signal users to interact with it, and were concerned that some attributes would have little meaning to those without music training.

Participant D was interested in exploring features that would make use of Spotify's playlist functionality. They shared that they were interested in using **DalSegno** to make a playlist for a party, and thus wondered if it would be possible to consider data from other users' profiles or playlists to generate playlists that had songs several people would like.

5.1.3 Discussion

The alpha tests revealed valuable insight into users' perspectives on an early prototype of **DalSegno**. Overall, participants found the **DalSegno** experience enjoyable and engaging. Users enjoyed driving their self-exploration of music, actively seeking out songs from new genres and artists. They also frequently leveraged the attribute graphs to request songs with specific characteristics, and found it engaging to test how different parameters could influence their perceptions of music. This demonstrates that providing functionality empowering users to discover music aligned with their preferences was a pivotal aspect of their satisfaction and engagement with the system.

However, these tests also revealed several design failures. Some were relatively minor visual bugs that had quick fixes. Others were more difficult to address as they required the interface or recommendation algorithm to be redesigned. Users also indicated that they wished for closer integration with Spotify features.

To address the issues discovered through alpha testing, we made the following modifications to the prototype going forward:

- Instead of generating 8 recommendations per seed, limit the total amount of results to 10
- Limit track and artist title length to not overflow the table column widths
- Ensure that the same track is not recommended multiple times in a set of recommendations
- Generate suggestions for seeds based on the user's most-listened songs
- Move the icon that triggers the opening of a track's attribute graph to the left-hand side to make it more visible
- Add inline Spotify links to each recommendation so that users can listen to the track in their native Spotify app
- Improve the clustering algorithm to sort results by how closely each suggestion resembles its seed(s)
- Remove loudness and BPM from displayed song attributes

5.2 Beta Testing

As part of our beta testing phase, we orchestrated a series of unmoderated tests and instead made observations on how the platform was perceived and used through collecting site analytics and post-experience survey responses from the participants.

5.2.1 Methodology

Eight participants were recruited via word-of-mouth; a researcher reached out to several close friends who used Spotify and were willing to try the interface, which included individuals with backgrounds ranging from formal to no music training.

As a courtesy for our users' privacy, users were assigned a randomized site-generated ID to cite in their survey submissions. Additionally, specific song details were not collected; instead, site analytics trended towards tracking numerical counts of various interactions on the site:

- The number of rounds a user underwent during an experience
- The number of recommendations generated per round
- The total number of tracks saved as seeds during a round
- Whether a user listened to a song's audio preview, viewed its attribute graphs, or readjusted its attributes
- Amount of time a user spent on each page (in seconds)

The post-experience survey is structured to first elicit the user's experiences with Spotify and its recommendation features (Appendix B-2 through B-4). Afterwards, it delves into how each user perceived the **DalSegno** recommendation algorithm and interface design, before finally encouraging users to reflect on their overall experience.

5.2.2 Results

The beta tests revealed new insights about users' experiences on **DalSegno**. A total of 279 recommendations were generated for users that participated in a cumulative 34 rounds of recommendation (an average of 8.21 recommendations per round).

Users expressed universal acclaim for the visual design of the interface in the post-experience surveys (Figure 5-1). They were impressed by the aesthetic appeal and overall visual coherence of the system, which was attributed to colorful, interactive attribute graphs that complemented a simple, clean, and straightforward user interface design.

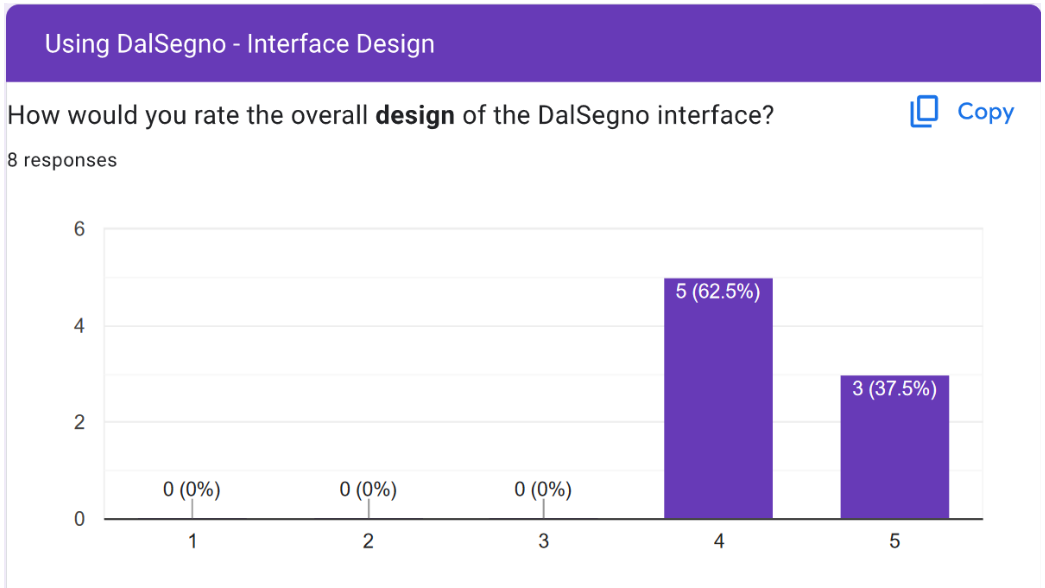


Figure 5-1: User ratings for the visual design of the beta prototype.

Figure 5-2 illustrates which features users interacted with as they navigated the interface. In particular, users generally listened to most tracks that they were suggested, and users rarely engaged with attribute graphs without first listening to a track's audio preview. This particular usage pattern surfaced by site analytics resonates with the survey findings that users considered audio previews the most important feature for exploring music preferences, while attribute graphs were rated highest for preference understanding. One user specifically remarked, "I think I can easily recognise if a certain song fits my preference, but it is hard for me to describe in words, so the application has been quite helpful." Several users expressed dissatisfaction with Spotify's ability to define their music preferences and identify common musical themes in their playlists, so such explainability was appreciated and contributed to users' willingness to further use the interface to explore their preferences.

The beta tests also revealed the potential for improvement in the interface's recommendation algorithm. Surprisingly, despite scoring high on similarity, the relevance of suggested tracks trended negatively as users underwent more rounds of recommen-

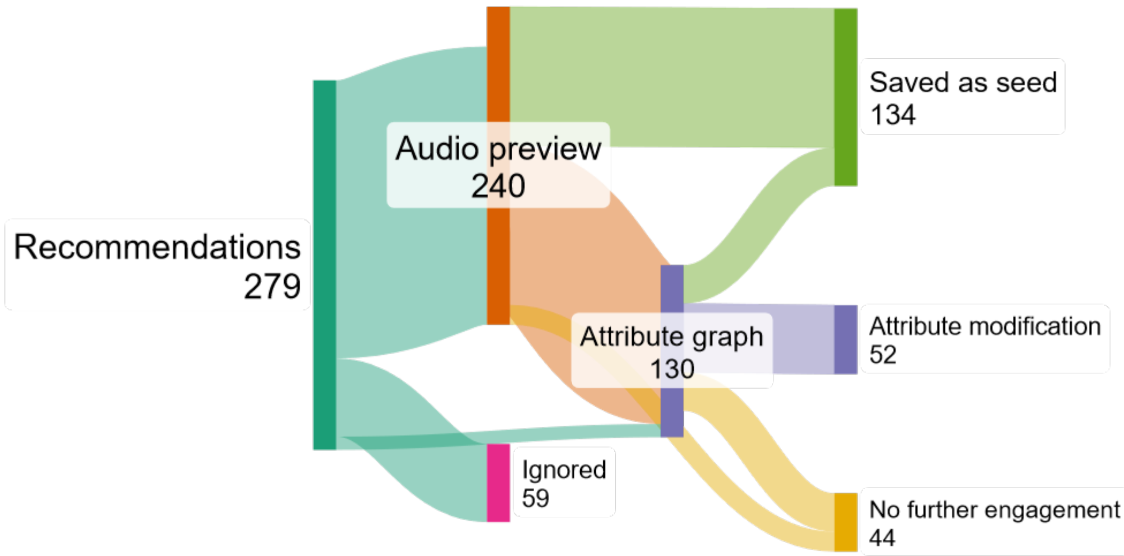


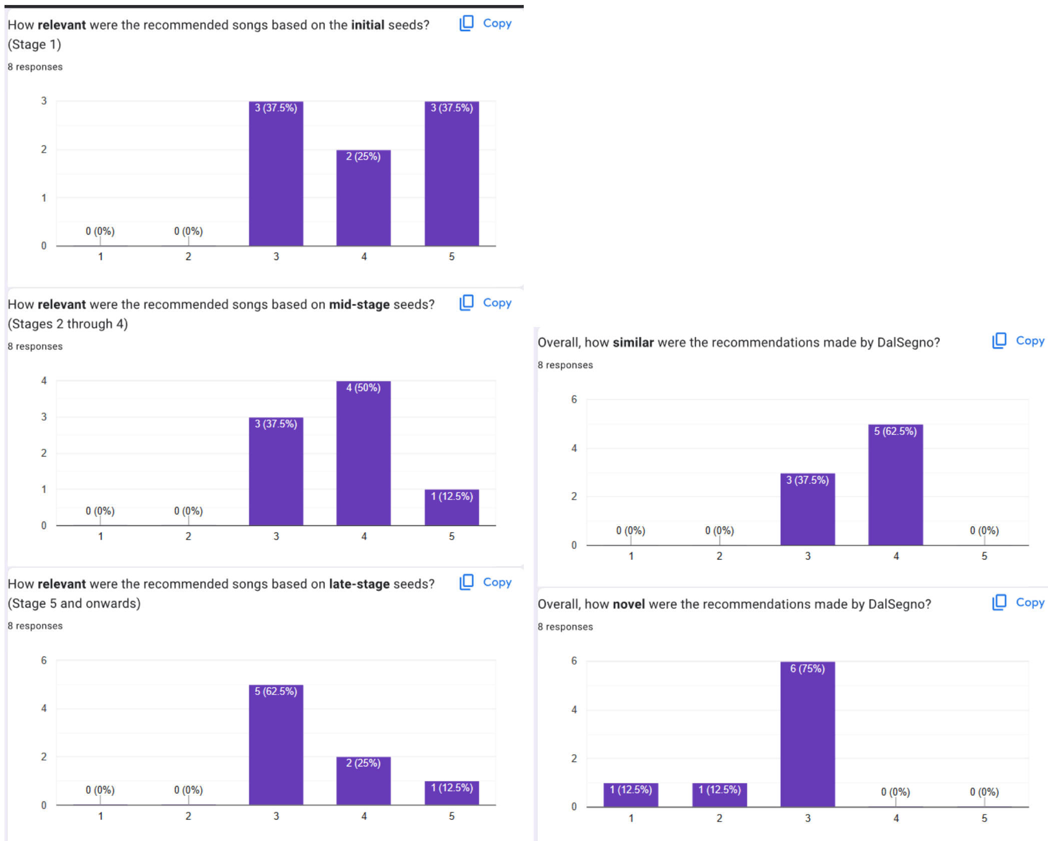
Figure 5-2: Progression of recorded user interactions during beta testing.

dations (Figure 5-3). Additionally, many users did not consider the recommendations novel, which indicates they consistently recommended several tracks that they already recognized.

The most cited critique for the interface was difficulty understanding the concept of seeds. Such feedback was not captured through the alpha tests, likely because the alpha testers all had extensive computer science education and would be familiar with how recommendation systems like **DalSegno** use seeds, while the beta users with more diverse backgrounds would not be so familiar. As a result, we had to consider how to better communicate the purpose of seeds to users.

One particular user echoed a feature request from Participant D in alpha testing in wanting to have seeds imported based on a playlist, complaining it was tedious to have to enter seeds one by one. Although developing features to "profile" a playlist was highly appealing, we were concerned about how the algorithm would scale to excessively large playlists or playlists with a mishmash of musical profiles. Thus, we did not move forward with the suggestion.

Finally, several users wrote that they found **DalSegno** more potent than existing



(a) Relevance ratings for recommendations.

(b) Similarity and novelty ratings for recommendations.

Figure 5-3: Beta test survey ratings of recommendation effectiveness.

Spotify recommendation features, such as Discover Weekly, Spotify Wrapped, and playlist-based suggestions. This implies that the strategies employed by **DalSegno** enable users to more quickly discover songs they like and understand their preferences, thus mitigating user cold start.

A compendium of survey responses from beta testing is included in Appendix B-7 through B-9.

5.2.3 Discussion

Given the above pain points, we implemented the following improvements for the final release of **DalSegno**:

- Initialize a series of interactive popups that appear on the seed selection and recommendations pages to guide the user around the interface and provide context to specific elements

- Limit the number of recommendations associated with each seed to ensure equal representation of seeds in the set of outputted recommendations
- Modify the recommendation algorithm to separate input seeds into clusters (Section 4.3.4)
- Add export to Spotify playlist functionality for clusters of saved tracks
- Add detailed summary pages with textual descriptions of inferred preferences (as opposed to the previous implementation that regurgitated raw statistics)

5.3 Final Release

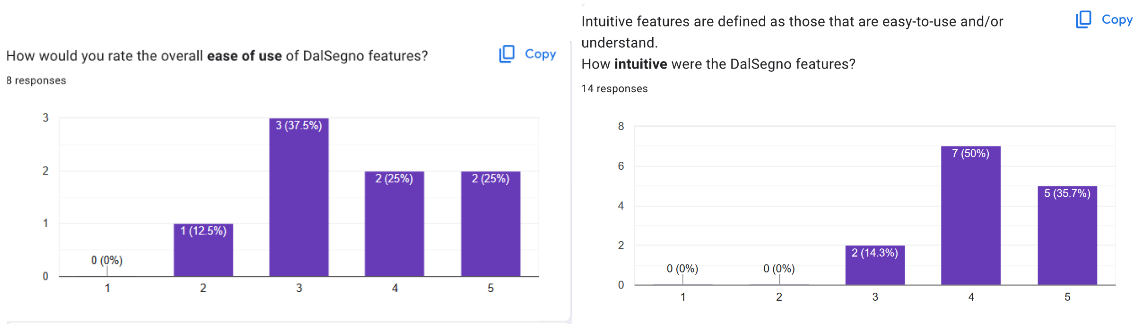
After improving the beta prototype, we asked our largest audience of users to test with the final prototype so that we could evaluate the impact of these enhancements on user experiences. Similar to beta tests, these tests were performed without moderation, and we relied on website analytics and post-experience survey responses to gauge usability.

5.3.1 Methodology

The methodology of the final release is highly similar to the strategy deployed for beta tests. Significant differences include recruiting a larger participant pool (11 new individuals, 3 returning participants) and a rewording/rearrangement of several survey questions (Appendix B-5 through B-6).

5.3.2 Results and Discussion

Overall, the modifications to the beta prototype resulted in tangible improvements for users. The addition of interactive popups to guide users around the seed and recommendation pages contributed to smoother navigation, and users also appreciated the clarity and informativeness provided by the popups. As a result, participants expressed increased satisfaction with the intuitiveness of the user interface (Figure 5-4).

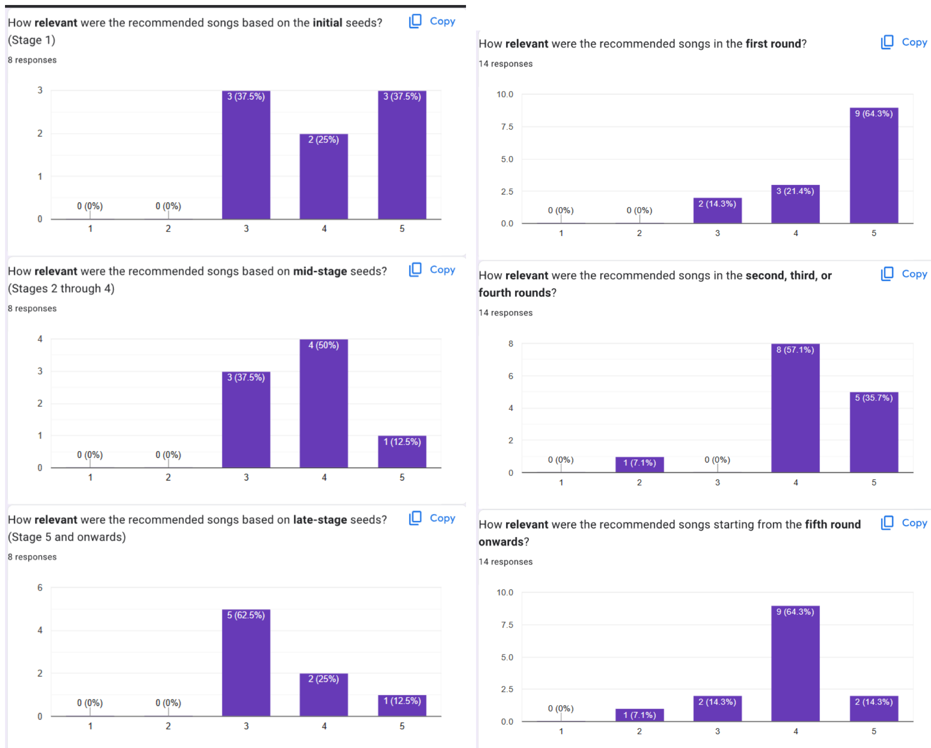


(a) Beta ratings for interface ease-of-use. (b) Final ratings for interface intuitiveness.

Figure 5-4: Comparison of user ratings on interface usability across beta and final survey results.

Another finding was that users perceived the recommendations had become more relevant and attuned to their preferences (Figure 5-5). Being able to discern disparate groups of musical preferences within the broader selection of input seeds provided a more finely tailored set of recommendations; users similarly appreciated that all seeds were represented in each set of outputs. The improved relevance in recommendations had a positive impact on user engagement with the system: analytics demonstrated that users spent more time exploring and interacting with the recommended songs in final tests than in beta tests (Figure 5-6).

One point of feedback that persisted throughout the beta and final tests was that users did not necessarily consider a majority of suggestions novel (Figure 5-7). Multiple users reported seeing suggestions recurring throughout several rounds of recommendations (though fortunately not within the same round), and some noted that they recognized suggestions more often than not. We hypothesize that this phenomenon occurs because Spotify prioritizes recommending tracks that appear in a user's previous listening history. However, it would be difficult to scale a system to scan through all of a user's playlists to avoid recommending a track a user has already encountered. On the other hand, some users appreciated seeing familiar recommendations, noting that it indicated the system was well-aware of their unique



(a) Beta relevance ratings for recommendations.

(b) Final relevance ratings for recommendations.

Figure 5-5: Comparison of user ratings on recommendation relevance across beta and final survey results.

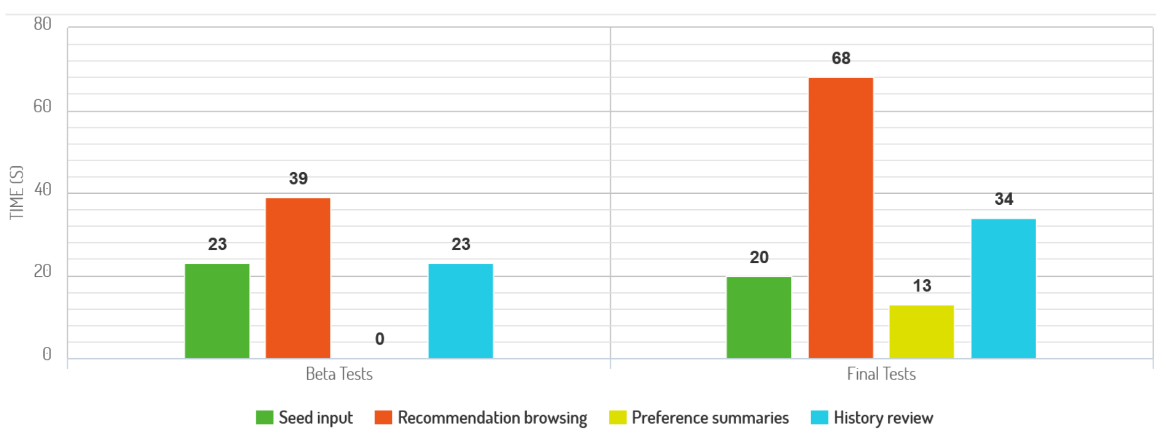
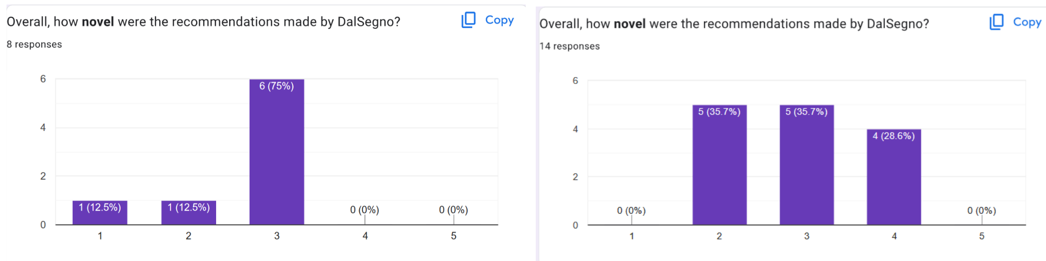


Figure 5-6: Average time spent on each page by users per session.

preferences.

The addition of preference summaries did not seem to have much of a positive impact on the user experience. Although one user appreciated that it helped "ground" or remind them of how their experience had transpired thus far, another commented



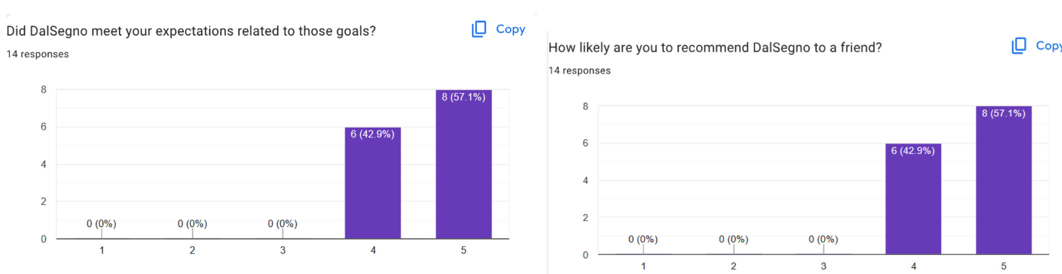
(a) Beta novelty ratings for recommendations.

(b) Final novelty ratings for recommendations.

Figure 5-7: Comparison of user ratings on recommendation novelty across beta and final survey results.

that it was redundant with the history review page. Analytics showed that users barely interacted with the summary (Figure 5-6), with some even skipping it to more quickly advance to their next sets of recommendations. We suspect a better way to approach preference summaries would be to allow users to generate them on demand (similar to how track history review is implemented now), and eliminate overlap between the two features.

Regardless, users continued to express satisfaction with their **DalSegno** experiences and felt that they were highly rewarding (Figure 5-8). A compendium of final test survey responses is included in Appendix B-10 through B-12.



(a) User ratings for interface effectiveness.

(b) User endorsement ratings.

Figure 5-8: Metrics of user satisfaction on the final prototype.

Chapter 6

Conclusion

Similar to how **DalSegno** experiences are structured, iteratively eliciting and incorporating user feedback from multiple stages of user testing allowed us to surface several critical insights. First, we discovered that identifying trends in seeds allows recommendation algorithms to find items that more closely capture disparate preferences, rather than items that combine characteristics in unexpected and undesirable ways. Next, we found that cognitive affordances, such as interactive popups and label definitions, are integral to helping users understand how to effectively navigate and utilize interfaces, no matter how redundant they may initially seem. Additionally, we learned that users enjoy driving their own user experiences when given opportunities to do so, but struggle when presented with too many options to choose from. Most importantly, we found that user cold start was mitigated by allowing users to personally influence recommendations, which helped them to discover favorable suggestions despite lacking an extensive listening history.

6.1 Future Work

Although **DalSegno** is already a thoroughly developed platform, we believe there are several other avenues we can explore to further evolve **DalSegno** into a more successful platform.

To encourage further user adoption, we would consider integrating more social sharing functions into **DalSegno**. For example, allowing users to compare attribute graphs and experiences through generated images or links could strike rapport among users with similar music preferences. On the other hand, some **DalSegno** testers requested being able to profile and use playlists as seeds so that they could gain a more comprehensive understanding of their music tastes; they remarked that if such a feature were implemented, they would be incentivized to use **DalSegno** daily.

From a developer's perspective, it would be ideal to move the **DalSegno** Spotify API client out of "development mode", as this limits the number of requests and user data that **DalSegno** can simultaneously access and would prevent **DalSegno** from scaling to a larger audience. However, two previous applications to remove such limitations were denied. As a result, it may be worth it to consider other less-restrictive sources for audio analysis, audio metadata, and recommendation data. Doing so would likely improve recommendation novelty, but may harm relevancy and similarity if the quality of data sources is not on par with Spotify.

In terms of user interface design, it would be interesting to see whether users would prefer a more guided approach to reviewing recommendations. Instead of manually clicking on songs to reveal their audio previews and attribute graphs, **DalSegno** would individually present users with all of the recommendation's data at once. Such a strategy may become cumbersome if they encounter a series of uninteresting suggestions and encourage them to become complacent in offering nuanced feedback, but we would need to first develop a prototype for this alternative interface to test this hypothesis.

Appendix A

Software Libraries

Package Name	Purpose
annoy https://github.com/spotify/annoy	Nearest-neighbor search implementation.
Flask https://flask.palletsprojects.com	Web framework used for implementing the backend.
pandas https://pandas.pydata.org/	Library providing utility methods for data analysis and manipulation.
python-dotenv https://pypi.org/project/python-dotenv	Library dedicated to importing .env configuration files into Python environments.
scikit-learn https://scikit-learn.org	Predictive data analysis library used for clustering.
spotipy https://spotipy.readthedocs.io/	Library for interacting with the Spotify web API.

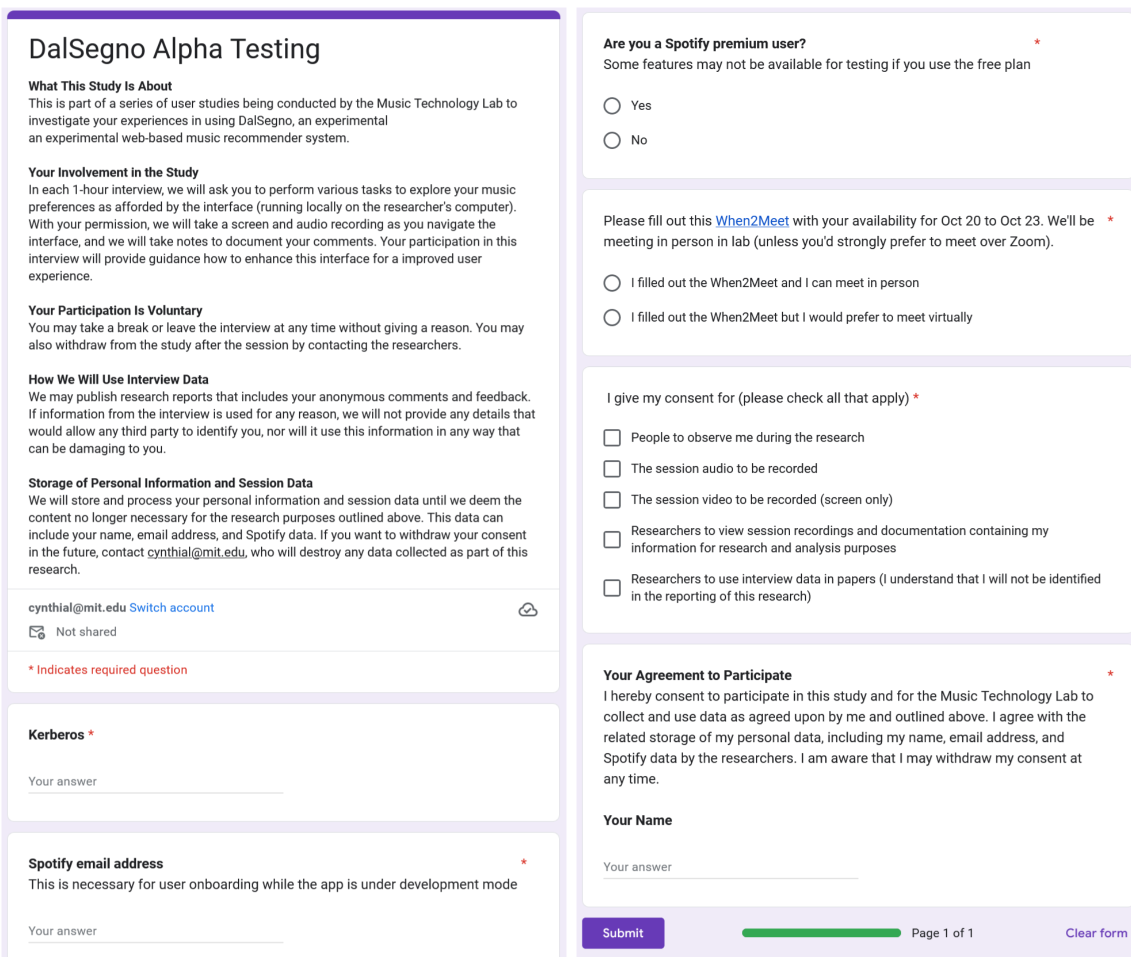
Figure A-1: Direct Python dependencies.

Package Name	**Purpose**
`naive-ui` https://naiveui.com	Component library that offers a set of pre-styled yet customizable UI components, such as text inputs, modals, tables, and dialogs.
`tailwindcss` https://tailwindcss.com	Set of pre-defined, shorthand utility classes for styling, which enables productivity and consistency throughout the project.
`unocss` https://unocss.dev	CSS postprocessor that dynamically eliminates unused styles, resulting in smaller and more optimized stylesheets.
`shepherd.js` https://github.com/synicalsyntax/shepherd	Component and logic library simplifying the creation of interactive walkthroughs to introduce site features and functionalities.
`@vueuse/sound` https://github.com/raffaelesgarro/vue-use-sound	Vue plugin simplifying control over audio playback on web browsers.
`chart.js` https://chartjs.org	High-level API and plugin ecosystem for creating custom interactive charts.
`chartjs-plugin-dragdata` https://github.com/synicalsyntax/chartjs-plugin-dragdata	`chart.js` plugin implementing support for draggable nodes on interactive charts.
`unplugin` https://github.com/unplugin	Collection of utilities streamlining the development of Vue applications, such as auto-importing components, icon sets, and modules.
`vue-router` https://router.vuejs.org	Vue's official routing package for implementing single-page applications.
`vite` https://vitejs.dev	Tool that optimizes the development and production build generation of reactive web applications.
`vite-plugin-pages` https://npmjs.com/package/vite-plugin-pages	Vite build system plugin for the dynamic generation of routes based on file directory structure.
`pinia` https://pinia.vuejs.org	Implements session-based state management for Vue applications, which can be configured with plugins.

Figure A-2: Direct JavaScript dependencies.

Appendix B

User Testing Resources



DalSegno Alpha Testing

What This Study Is About
This is part of a series of user studies being conducted by the Music Technology Lab to investigate your experiences in using DalSegno, an experimental
an experimental web-based music recommender system.

Your Involvement in the Study
In each 1-hour interview, we will ask you to perform various tasks to explore your music preferences as afforded by the interface (running locally on the researcher's computer). With your permission, we will take a screen and audio recording as you navigate the interface, and we will take notes to document your comments. Your participation in this interview will provide guidance how to enhance this interface for a improved user experience.

Your Participation Is Voluntary
You may take a break or leave the interview at any time without giving a reason. You may also withdraw from the study after the session by contacting the researchers.

How We Will Use Interview Data
We may publish research reports that includes your anonymous comments and feedback. If information from the interview is used for any reason, we will not provide any details that would allow any third party to identify you, nor will it use this information in any way that can be damaging to you.

Storage of Personal Information and Session Data
We will store and process your personal information and session data until we deem the content no longer necessary for the research purposes outlined above. This data can include your name, email address, and Spotify data. If you want to withdraw your consent in the future, contact cynthial@mit.edu, who will destroy any data collected as part of this research.

cynthial@mit.edu Switch account
Not shared

* Indicates required question

Kerberos *
Your answer

Spotify email address *
This is necessary for user onboarding while the app is under development mode
Your answer

Are you a Spotify premium user? *
Some features may not be available for testing if you use the free plan

- Yes
- No

Please fill out this When2Meet with your availability for Oct 20 to Oct 23. We'll be meeting in person in lab (unless you'd strongly prefer to meet over Zoom). *

- I filled out the When2Meet and I can meet in person
- I filled out the When2Meet but I would prefer to meet virtually

I give my consent for (please check all that apply) *

- People to observe me during the research
- The session audio to be recorded
- The session video to be recorded (screen only)
- Researchers to view session recordings and documentation containing my information for research and analysis purposes
- Researchers to use interview data in papers (I understand that I will not be identified in the reporting of this research)

Your Agreement to Participate *
I hereby consent to participate in this study and for the Music Technology Lab to collect and use data as agreed upon by me and outlined above. I agree with the related storage of my personal data, including my name, email address, and Spotify data by the researchers. I am aware that I may withdraw my consent at any time.

Your Name
Your answer

Submit — Page 1 of 1 — Clear form

Figure B-1: Participant recruitment form for alpha testing.

DalSegno User Experience Survey

Thank you for participating in the **DalSegno User Experience Survey**! We sincerely appreciate your willingness to share your thoughts. Through this survey, we hope to gather any feedback that you provide so that we can gather insights into how to better enhance our system.

Your responses will be anonymized and aggregated with others to analyze trends and areas for improvement; individual excerpts or summaries may be included in academic publications for research purposes. If you have any questions or concerns about this survey, please reach out to cynthial@mit.edu.

Thank you for your time!

* Indicates required question

1. Name

2. User ID from History page *

Your Background

In this section, we'd like to learn more about your music streaming habits on Spotify so that we can better understand your perspective on DalSegno.

3. Roughly, how long have you been using Spotify?

Mark only one oval.

- I have used Spotify for multiple years.
- I have used Spotify for around a year.
- I have used Spotify for a few months or less.

4. In general, how often do you use Spotify?

Mark only one oval.

- Consistently - on a near-daily basis.
- Frequently - at least three days a week.
- Sometimes - at least once a week.
- Rarely - once a month or rarer.

5. Would you consider Spotify your main way to listen to music?

Mark only one oval.

- Yes, it's my primary music streaming service.
- Yes, but I sometimes use other platforms to listen to music as well.
- No, I prefer using other music platforms to listen to music.
- No, I don't stream or listen to music often.

6. Is your Spotify account registered on a Premium plan?

Mark only one oval.

- Yes
- No

7. Which of the following Spotify music recommendation features do you use?

Check all that apply.

- Discover Weekly
- Release Radar
- Daily Mixes
- Genre/artist-based mixes
- None of the above
- Other:

8. How would you describe your satisfaction with Spotify's music recommendation features?

Mark only one oval.

	1	2	3	4	5	
Very unsatisfied						Highly satisfied

9. What features or improvements would you like Spotify to implement to better enable you to discover new music?

10. Which of the following Spotify features do you use to understand your music preferences?

Check all that apply.

- Spotify Wrapped
- Top artists/tracks this month (displayed on your user profile)
- Genre/artist-based mixes
- Third-party Spotify integrations like Spotistats/stats.fm and last.fm
- None of the above
- Other:

11. What features or improvements would you like Spotify to implement to better enable you to understand your music preferences?

Using DalSegno - Recommendation Effectiveness

12. Describe your goals when you started using DalSegno as a new user. Were you hoping to discover new artists or genres, or find similar songs to what you enjoy? Were you trying to learn about your music preferences?

13. Did DalSegno meet your expectations related to those goals?

Mark only one oval.

	1	2	3	4	5	
Not at all						Exceeded expectations

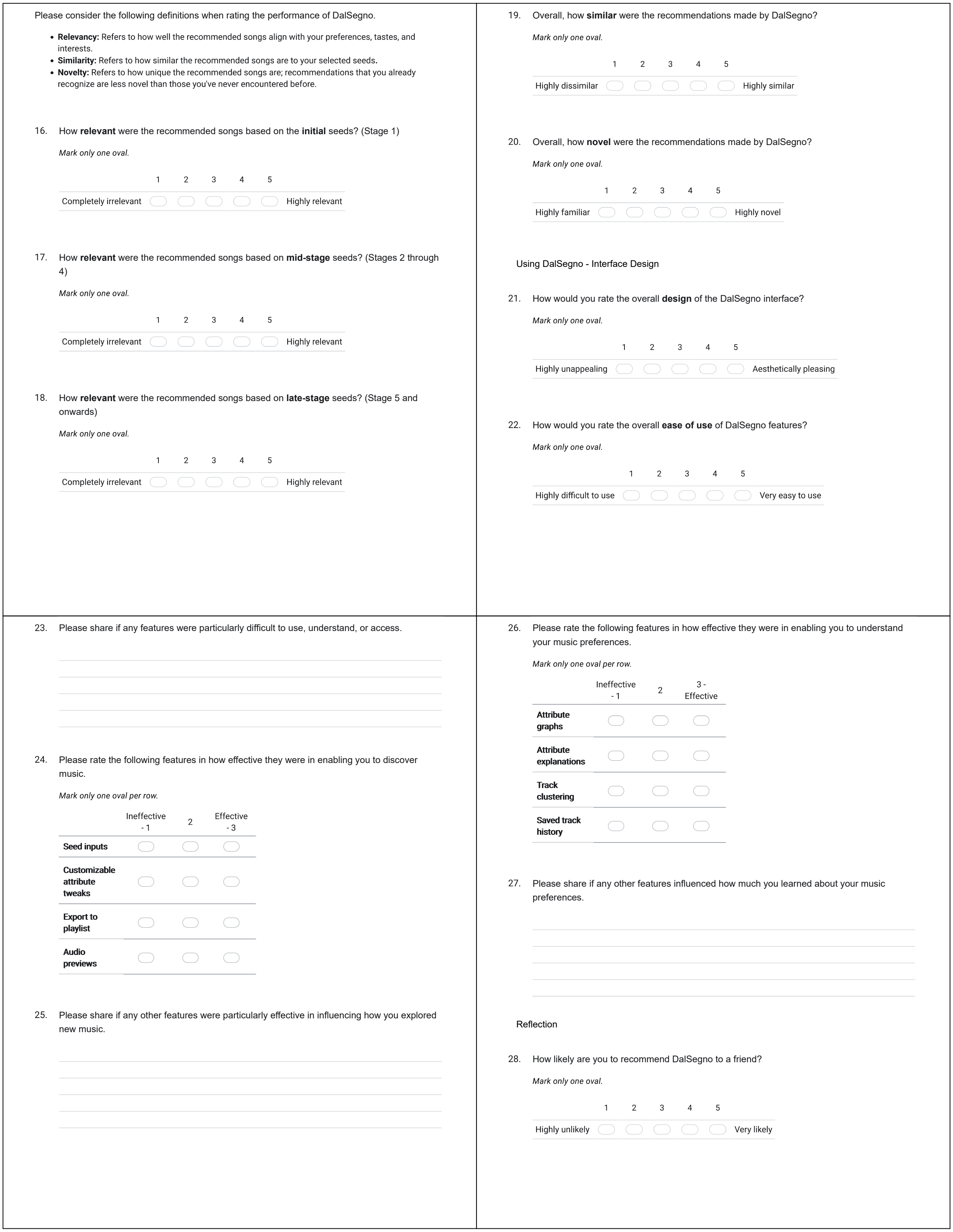
14. Please elaborate on your answer to the previous question.

15. Which strategy best describes how you chose your initial seeds for your Experience?

Check all that apply.

- Chose a single initial seed that I wanted to find songs most similar to
- Chose multiple seeds of a similar genre or style
- Chose multiple seeds of varying genres or style
- Chose seed(s) based on top tracks in my recent listening history
- Other:

Figure B-2: User experience survey for beta testing (Part 1)



Please consider the following definitions when rating the performance of DalSegno.

- **Relevancy:** Refers to how well the recommended songs align with your preferences, tastes, and interests.
- **Similarity:** Refers to how similar the recommended songs are to your selected seeds.
- **Novelty:** Refers to how unique the recommended songs are; recommendations that you already recognize are less novel than those you've never encountered before.

16. How **relevant** were the recommended songs based on the **initial** seeds? (Stage 1)

Mark only one oval.

	1	2	3	4	5	
Completely irrelevant						Highly relevant

17. How **relevant** were the recommended songs based on **mid-stage** seeds? (Stages 2 through 4)

Mark only one oval.

	1	2	3	4	5	
Completely irrelevant						Highly relevant

18. How **relevant** were the recommended songs based on **late-stage** seeds? (Stage 5 and onwards)

Mark only one oval.

	1	2	3	4	5	
Completely irrelevant						Highly relevant

19. Overall, how **similar** were the recommendations made by DalSegno?

Mark only one oval.

	1	2	3	4	5	
Highly dissimilar						Highly similar

20. Overall, how **novel** were the recommendations made by DalSegno?

Mark only one oval.

	1	2	3	4	5	
Highly familiar						Highly novel

Using DalSegno - Interface Design

21. How would you rate the overall **design** of the DalSegno interface?

Mark only one oval.

	1	2	3	4	5	
Highly unappealing						Aesthetically pleasing

22. How would you rate the overall **ease of use** of DalSegno features?

Mark only one oval.

	1	2	3	4	5	
Highly difficult to use						Very easy to use

23. Please share if any features were particularly difficult to use, understand, or access.

24. Please rate the following features in how effective they were in enabling you to discover music.

Mark only one oval per row.

	Ineffective - 1	2	Effective - 3
Seed inputs			
Customizable attribute tweaks			
Export to playlist			
Audio previews			

25. Please share if any other features were particularly effective in influencing how you explored new music.

26. Please rate the following features in how effective they were in enabling you to understand your music preferences.

Mark only one oval per row.

	Ineffective - 1	2	3 - Effective
Attribute graphs			
Attribute explanations			
Track clustering			
Saved track history			

27. Please share if any other features influenced how much you learned about your music preferences.

Reflection

28. How likely are you to recommend DalSegno to a friend?

Mark only one oval.

	1	2	3	4	5	
Highly unlikely						Very likely

Figure B-3: User experience survey for beta testing (Part 2)

29. What do you like most about the current DalSegno interface?

30. What improvements would you suggest to enhance the DalSegno experience?

31. Are there any specific features or functionalities you wish DalSegno had?

32. How does DalSegno compare to other music recommendation platforms you've used?

33. Please share any remaining thoughts you might have about DalSegno.

This content is neither created nor endorsed by Google.

Google Forms

Figure B-4: User experience survey for beta testing (Part 3)

DalSegno User Experience Survey

Thank you for participating in the **DalSegno User Experience Survey**! We sincerely appreciate your willingness to share your thoughts. Through this survey, we hope to gather any feedback that you provide so that we can gather insights into how to better enhance our system.

Your responses will be anonymized and aggregated with others to analyze trends and areas for improvement; individual excerpts or summaries may be included in academic publications for research purposes. If you have any questions or concerns about this survey, please reach out to cynthial@mit.edu.

Thank you for your time!

1. Name

Using DalSegno - Recommendation Effectiveness

2. Describe your goals when you started using DalSegno as a new user. Were you hoping to discover new artists or genres, or find similar songs to what you enjoy? Were you trying to learn about your music preferences?

3. Did DalSegno meet your expectations related to those goals?

Mark only one oval.

	1	2	3	4	5	
Not at all	◯	◯	◯	◯	◯	Exceeded expectations

4. Please elaborate on your answer to the previous question.

5. Which strategy best describes how you chose your initial seeds?

Check all that apply.

- ☐ Chose a single initial seed that I wanted to find songs most similar to
- ☐ Chose multiple seeds of a similar genre or style
- ☐ Chose multiple seeds of different genres or styles
- ☐ Chose seed(s) based on top tracks in my recent listening history
- ☐ Other: ____________

Please consider the following definitions when rating the performance of DalSegno.

- **Relevancy:** Refers to how well the recommended songs align with your preferences, tastes, and interests.
- **Similarity:** Refers to how similar the recommended songs are to your selected seeds.
- **Novelty:** Refers to how unique the recommended songs are; recommendations that you already recognize are less novel than those you've never encountered before.

6. How **relevant** were the recommended songs in the **first round**?

Mark only one oval.

	1	2	3	4	5	
Completely irrelevant	◯	◯	◯	◯	◯	Highly relevant

7. How **relevant** were the recommended songs in the **second, third, or fourth rounds**?

Mark only one oval.

	1	2	3	4	5	
Completely irrelevant	◯	◯	◯	◯	◯	Highly relevant

8. How **relevant** were the recommended songs starting from the **fifth round onwards**?

Mark only one oval.

	1	2	3	4	5	
Completely irrelevant	◯	◯	◯	◯	◯	Highly relevant

9. Overall, how **similar** were the recommendations made by DalSegno?

Mark only one oval.

	1	2	3	4	5	
Highly dissimilar	◯	◯	◯	◯	◯	Highly similar

10. Overall, how **novel** were the recommendations made by DalSegno?

Mark only one oval.

	1	2	3	4	5	
Highly familiar	◯	◯	◯	◯	◯	Highly novel

11. Please expand on your ratings.

Using DalSegno - Interface Design

12. How would **visually appealing** was the **design** of the DalSegno interface?

Mark only one oval.

	1	2	3	4	5	
Highly unappealing	◯	◯	◯	◯	◯	Aesthetically pleasing

13. Intuitive features are defined as those that are easy-to-use and/or understand. How **intuitive** were the DalSegno features?

Mark only one oval.

	1	2	3	4	5	
Difficult to use and understand	◯	◯	◯	◯	◯	Highly intuitive

14. Please share if any features were particularly difficult to use or understand.

Figure B-5: User experience survey for final testing (Part 1)

15. Please rate the following features in how effective they were in enabling you to discover music.

Mark only one oval per row.

	Ineffective - 1	2	Effective - 3	I did not use or encounter this feature
Seed inputs	◯	◯	◯	◯
Customizable attribute tweaks	◯	◯	◯	◯
Export to playlist	◯	◯	◯	◯
Audio previews	◯	◯	◯	◯

16. Please share if any other features were particularly effective in influencing how you explored new music.

17. Please rate the following features in how effective they were in enabling you to understand your music preferences.

Mark only one oval per row.

	Ineffective - 1	2	3 - Effective	I did not use or encounter this feature
Attribute graphs	◯	◯	◯	◯
Attribute explanations	◯	◯	◯	◯
Track clustering	◯	◯	◯	◯
Saved track history	◯	◯	◯	◯
Preference summaries	◯	◯	◯	◯

18. Please share if any other features influenced how much you learned about your music preferences.

Reflection

19. How likely are you to recommend DalSegno to a friend?

Mark only one oval.

	1	2	3	4	5	
Highly unlikely	◯	◯	◯	◯	◯	Very likely

20. What do you like most about the current DalSegno interface?

21. What improvements would you suggest to enhance the DalSegno experience?

22. Are there any specific features or functionalities you wish DalSegno had?

23. How does DalSegno compare to other music recommendation platforms or features you've used?
Examples: Spotify Wrapped, Discover Weekly, Release Radar, Daily Mixes

24. Please share any remaining thoughts you may have about DalSegno.

This content is neither created nor endorsed by Google.

Google Forms

Figure B-6: User experience survey for final testing (Part 2)

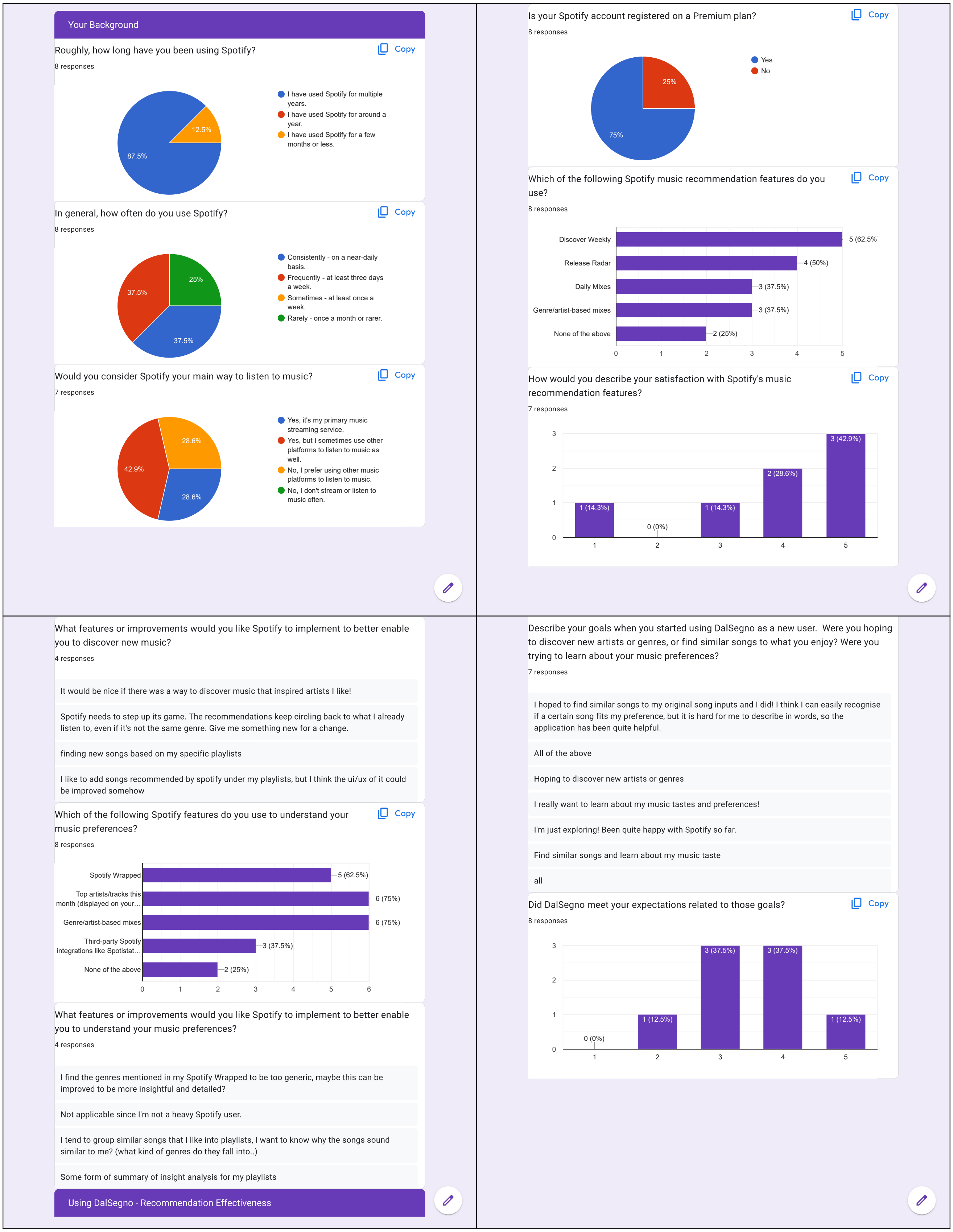


Figure B-7: Survey results from beta tests (Part 1)

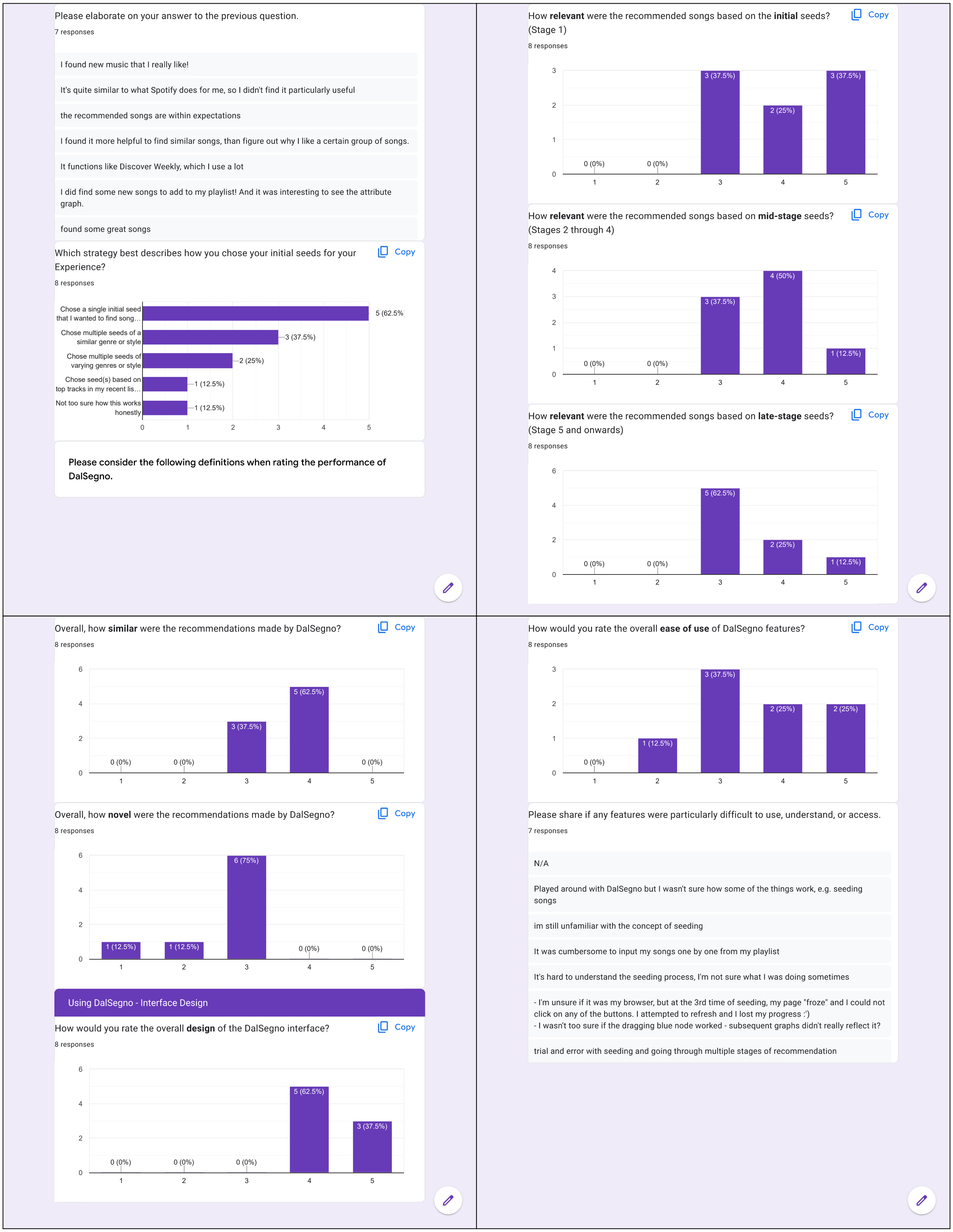


Figure B-8: Survey results from beta tests (Part 2)

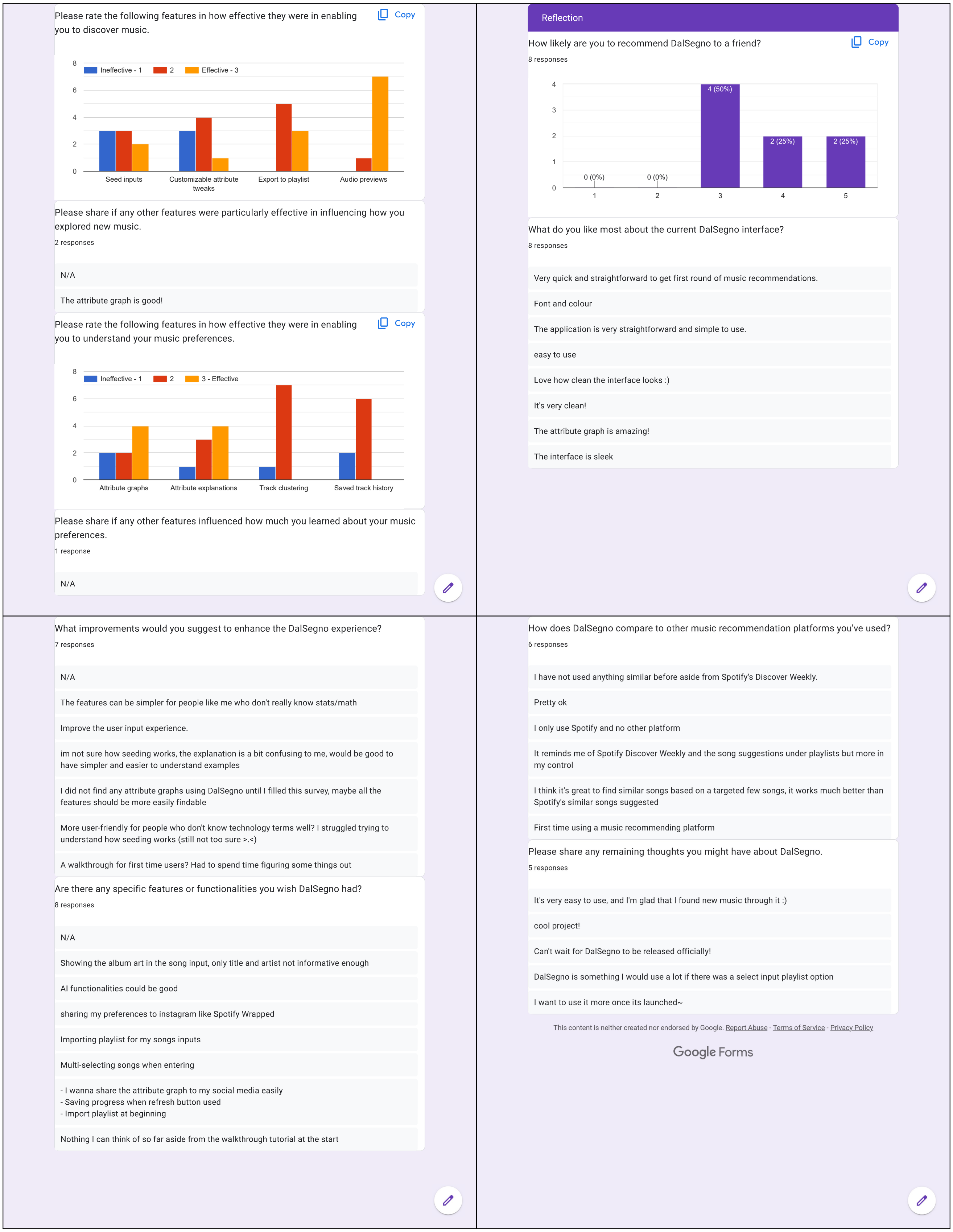


Figure B-9: Survey results from beta tests (Part 3)

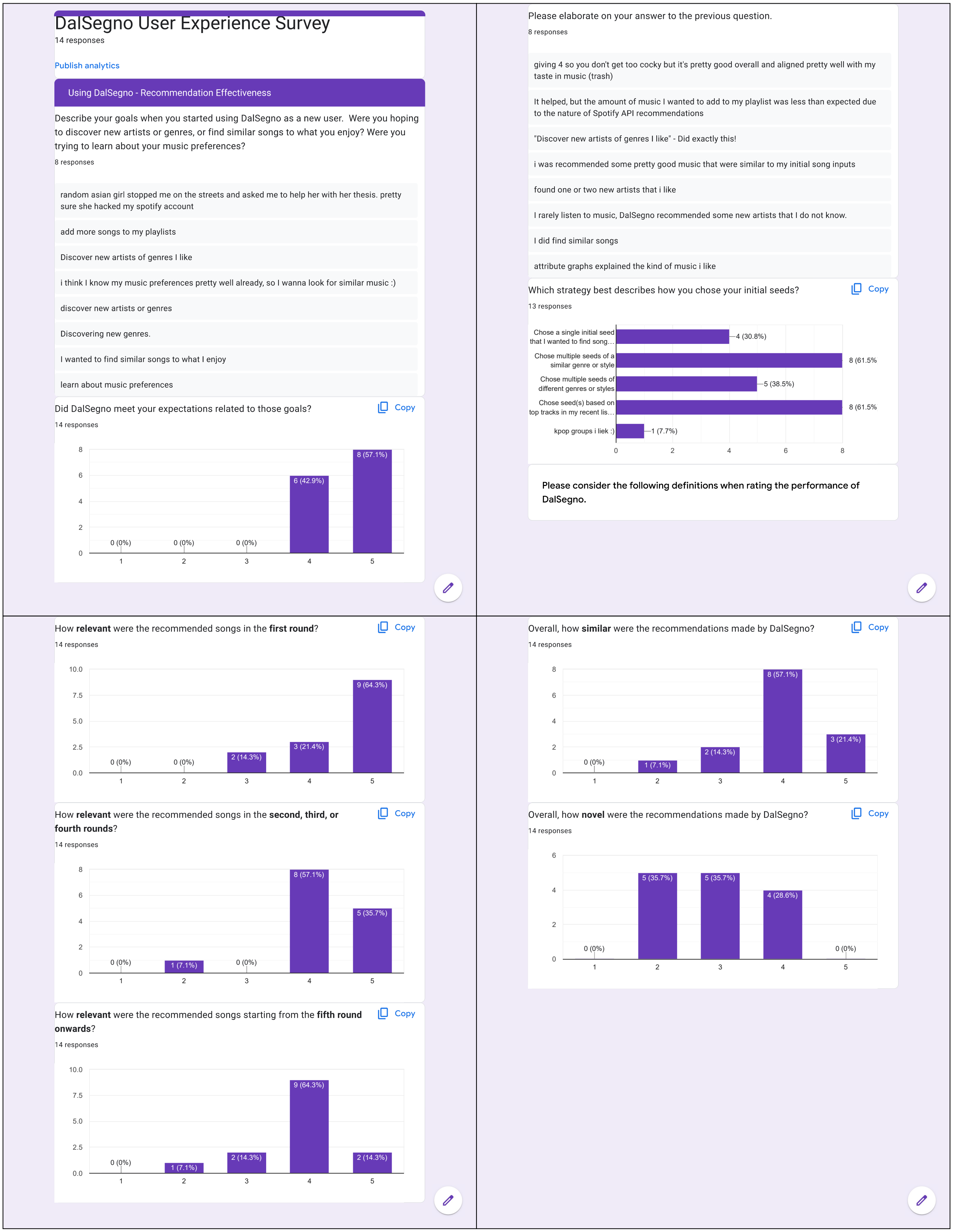


Figure B-10: Survey results from final tests (Part 1)

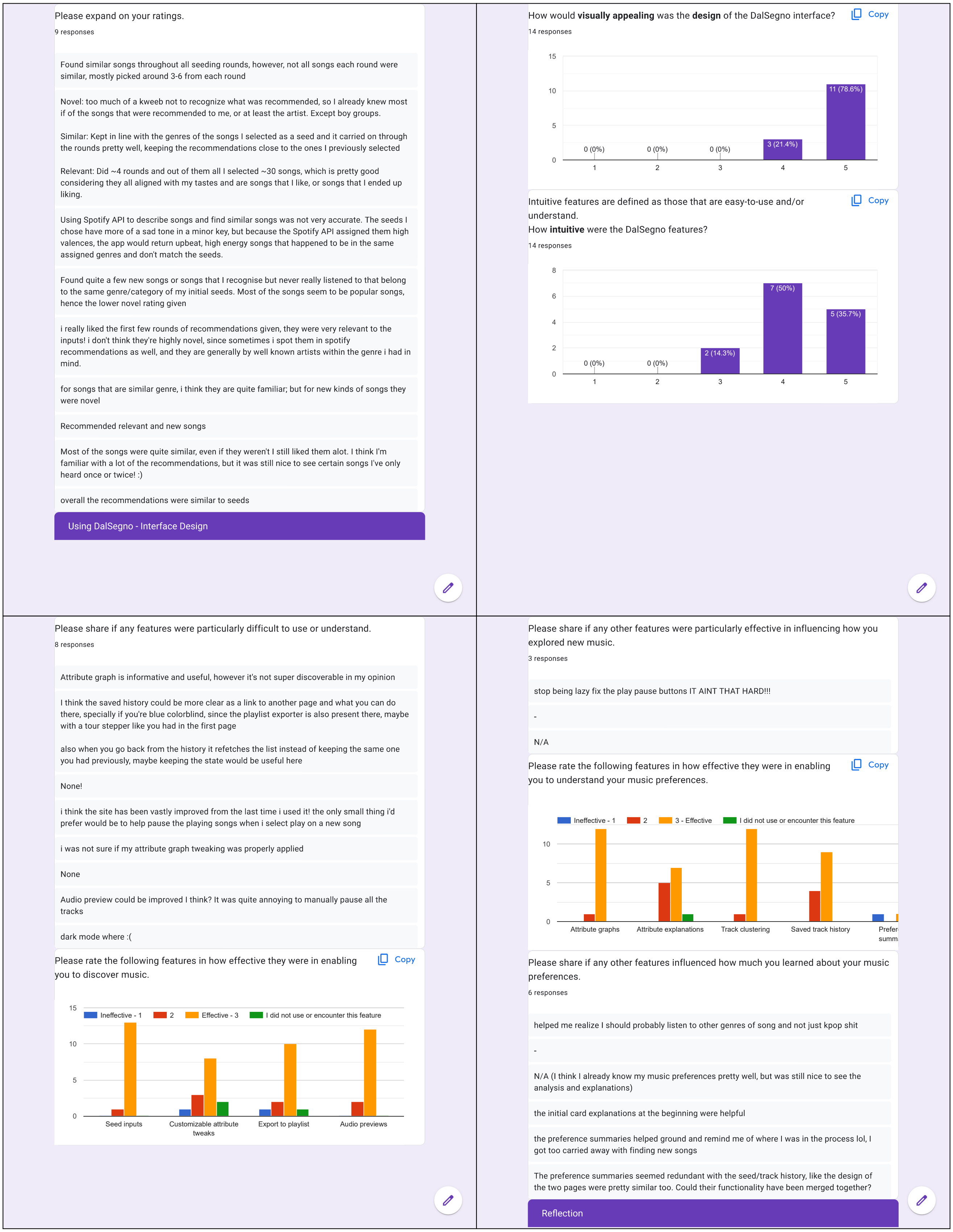
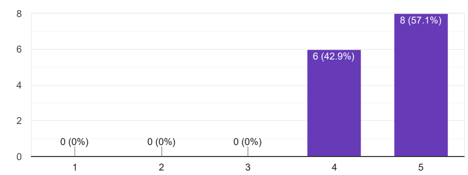


Figure B-11: Survey results from final tests (Part 2)

How likely are you to recommend DalSegno to a friend?

14 responses



What do you like most about the current DalSegno interface?

9 responses

- Super simple and pretty and easy to use. I also really like the export to playlist option!
- Cleanliness and ease of use. Maybe clicking on the row to select the track instead of having to specifically click on the checkbox would be helpful too. Also a link to the artist's page (the song link is already pretty useful however)
- The songs that the recommendation was influenced by and the overall attribute graph
- The interface is really clean, I like the export to playlist feature the most.
- The flow of usage is very intuitive and well guided now, I like how easy to use it is
- easy to use
- The interface is very clean and simple to understand
- The site is very consistent and I like how I could enter romaji versions of chinese/japanese songs and they'll still show up!
- minimalistic design, no ads

What improvements would you suggest to enhance the DalSegno experience?

9 responses

- Auto-pause playing songs when another song is clicked
- left in the other feedbacks
- Use seeds from the initial search (if there were multiple) in the second round. If only one song is selected in the first round, DalSegno only shows you music similar to that song.
- Restrict one song playing at a time, it gets quite noisy and annoying to pause multiple songs.
- Auto-pause songs playing when new song is selected
- have a notification or some cue that the attribution tweaks are applied
- None
- Improve the audio preview feature to only allow 1 playing song
- more kinds of attributions?

Are there any specific features or functionalities you wish DalSegno had?

9 responses

- Sharing the attribute graph to my instagram story :D
- I think it does what it means to do pretty well
- Automatically pause a track preview when you start listening to another track.
- Adding a filter for max number of listens a song can have? So I can find more "unconventional" and unknown music.
- Allow filtering by artists so that I can only filter for female singers; sorting by popularity
- can't think of anything new
- None
- It's good enough :D
- so far so good

How does DalSegno compare to other music recommendation platforms or features you've used?
Examples: Spotify Wrapped, Discover Weekly, Release Radar, Daily Mixes

9 responses

- I use Discover Weekly frequently, I think they are quite similar, but DalSegno allows more flexibility and customising options for me to filter further.
- better than the daily mixes although both of them keeps recommending me boygroups
- More useful than discovery weekly because at least i won't be forced to listen to lound noise flops without pulling out the official spotify app and trying to remove the song from the weekly playlist.
- I only used Spotify Wrapped, I think its much cooler because I find new music!
- I think DalSegno stands out from the examples because we're allowed to further customise and control the recommendations, instead of just having a single list of songs.
- good to find new artists/genre compared to the examples
- I don't use Spotify so I cannot really compare, but I think DalSegno is great!
- I think its more interactive and interesting than the ones mentioned.
- dalsegno and spotify wrapped both explain my song taste, dalsegno helped more by recommending other songs in the same genre rather than just providing a summary. i also found dalsegno to be more detailed

Please share any remaining thoughts you may have about DalSegno.

9 responses

- It's an amazing project :) All the best!
- very cool thesis you'll nail it
- put in peepy in the background
- Please add a feature or include songs that are much lesser known! Would use it much more frequently so I can discover small artists :)
- DalSegno (only female edition) please !! Very proud of Cynthia for this cool site that I'll be using in future to discover more shoegaze music :)
- good job!
- Cool application to find new songs
- Found a song I really liked, did not notice I have already saved in back in 2021. Thanks DalSegno for connecting me back to old music that I liked and forgot :)
- cool stuff~

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Privacy Policy

Google Forms

Figure B-12: Survey results from final tests (Part 3)

Bibliography

[1] Dmitry Bogdanov, Martín Haro, Ferdinand Fuhrmann, Anna Xambó, Emilia Gómez, and Perfecto Herrera. Semantic audio content-based music recommendation and visualization based on user preference examples. *Information Processing & Management*, 49(1):13–33, 2013.

[2] Léa Briand, Guillaume Salha-Galvan, Walid Bendada, Mathieu Morlon, and Viet-Anh Tran. A semi-personalized system for user cold start recommendation on music streaming apps. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2601–2609, 2021.

[3] Michael A Casey, Remco Veltkamp, Masataka Goto, Marc Leman, Christophe Rhodes, and Malcolm Slaney. Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, 96(4):668–696, 2008.

[4] Peter Knees, Markus Schedl, and Masataka Goto. Intelligent user interfaces for music discovery: The past 20 years and what's to come. In *ISMIR*, pages 44–53, 2019.

[5] Fang-Fei Kuo and Man-Kwan Shan. A personalized music filtering system based on melody style classification. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pages 649–652. IEEE, 2002.

[6] Rishabh Mehrotra, Chirag Shah, and Benjamin Carterette. Investigating listeners' responses to divergent recommendations. In *Fourteenth ACM Conference on Recommender Systems*, pages 692–696, 2020.

[7] Keisuke Okada, Phan Xuan Tan, and Eiji Kamioka. Five-factor musical preference prediction for solving new user cold-start problem in content-based music recommender system. In *2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA)*, pages 1–7. IEEE, 2021.

[8] Savvas Petridis, Nediyana Daskalova, Sarah Mennicken, Samuel F Way, Paul Lamere, and Jennifer Thom. Tastepaths: Enabling deeper exploration and understanding of personal preferences in recommender systems. In *27th International Conference on Intelligent User Interfaces*, pages 120–133, 2022.

[9] Sanjay Purushotham, Yan Liu, and C-C Jay Kuo. Collaborative topic regression with social matrix factorization for recommendation systems. *arXiv preprint arXiv:1206.4684*, 2012.

[10] Upendra Shardanand and Pattie Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217, 1995.

[11] Elizabeth S Spelke. Sex differences in intrinsic aptitude for mathematics and science?: a critical review. *American psychologist*, 60(9):950, 2005.

[12] Xinxi Wang, David Rosenblum, and Ye Wang. Context-aware mobile music recommendation for daily activities. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 99–108, 2012.

[13] Ali Yürekli, Alper Bilge, and Cihan Kaleli. Exploring playlist titles for cold-start music recommendation: an effectiveness analysis. *Journal of Ambient Intelligence and Humanized Computing*, 12(11):10125–10144, 2021.