

PROBLEM DESCRIPTION

Existing methods of sharing music leave a lot to be desired. Boring hyperlinks must be copied and pasted from native services into messaging applications to be shared, resulting in recipients missing or losing track of the recommendations they receive and senders feeling frustrated and annoyed when their efforts are not reciprocated. Although some music services like SoundCloud and Spotify have music sharing features, these features are restricted to users of the service and are still auxiliary to the core features of the platform. Even Shazam, which is designed to help you remember a song for later, still relies on links and screenshots for sharing.

Sharing music should be a frictionless process that is focused on the content being shared and the relationship development around this content rather than the logistics of sending content.

SOLUTION

The Musit team aims to streamline the music sharing process so that people can focus on sharing music and fostering relationships rather than on sending links. Music sharing should be about sharing with others rather than on logistics. We tackled this goal by creating a dedicated channel for sharing music with your friends and the people around you.

Musit streamlines the music-sharing process in two primary ways. First, Musit makes it easy to find and send a song as well as receive a song and listen to it through integration with Spotify and SoundCloud. Secondly, Musit organizes all music recommendations in one centralized location to prevent music recommendations from getting lost in the shuffle.

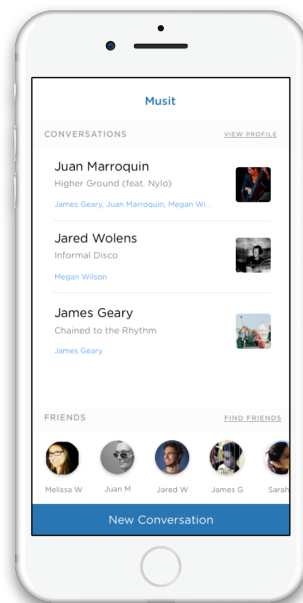


Figure 1: The Musit home screen, where users can view all of their music recommendations in one centralized location.

TASKS

At a fundamental level, sharing music is about sending and receiving recommendations. As such, Musit is designed to facilitate these core tasks.

Task 1: Sharing a Song Recommendation with an Individual (Simple)

The building blocks of a music sharing relationship are the individual song recommendations that are exchanged. Musit facilitates building music-based relationships by streamlining the task of sending a song recommendation to another individual.

Task #2: Receiving a Song Recommendation (Moderate)

Users need a convenient way to receive music recommendations from others. Musit streamlines this process by allowing users to listen a song that is recommended to them in Spotify or SoundCloud directly from the app, and by organizing all song recommendations in a centralized location.

Task #3: Sharing Music with a Group

Music naturally brings groups of individuals together, and Musit streamlines the process of building and maintaining music-centric communities. We decided to include the group sharing process in our application because many of the difficulties of sharing music with existing methods (sharing links in messaging apps, for example) are exacerbated when many people are contributing to the conversation; it is extremely difficult to keep track of who shares what and how different people feel about certain content. When members in a group admire another poster's content, they can go to that user's profile page and discover other songs that user has been sharing.

TASK FLOWS

Task 1: Sharing a Song Recommendation with an Individual (Simple)

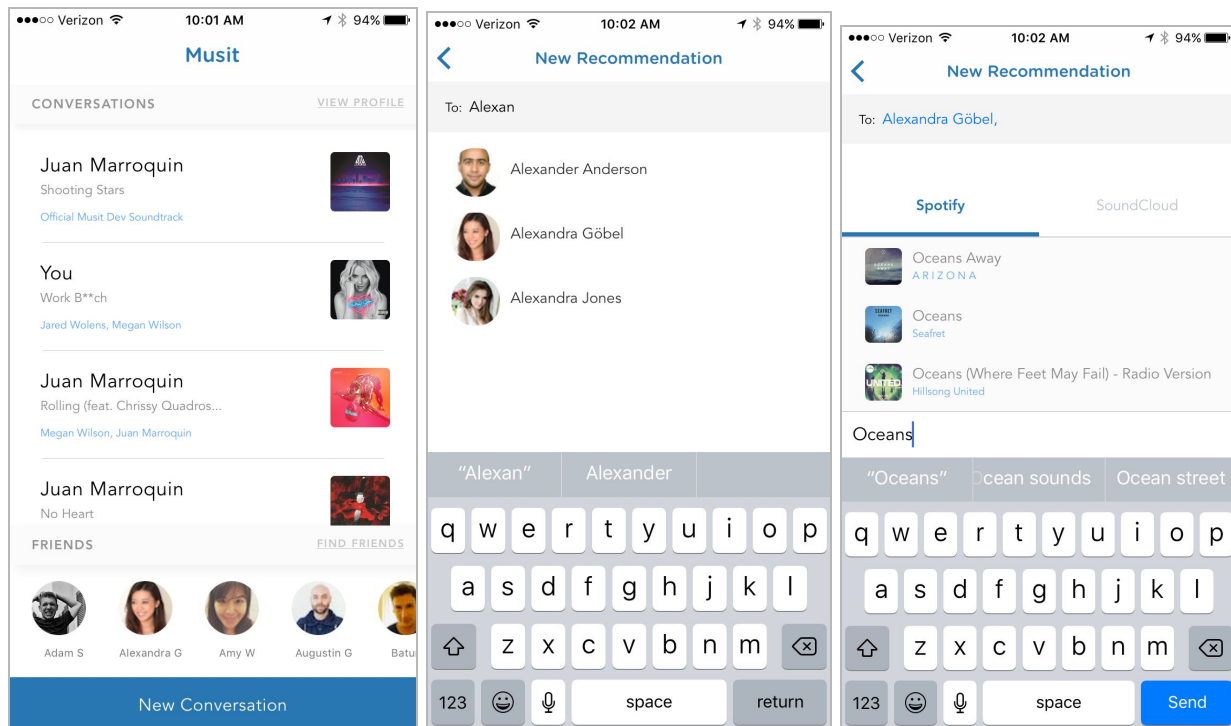


Figure 1: Start at the dashboard and click “new conversation” (left). Enter a friend’s name to send the recommendation to (center), then search for a song in Spotify or SoundCloud (toggle using the tab bars) (right).

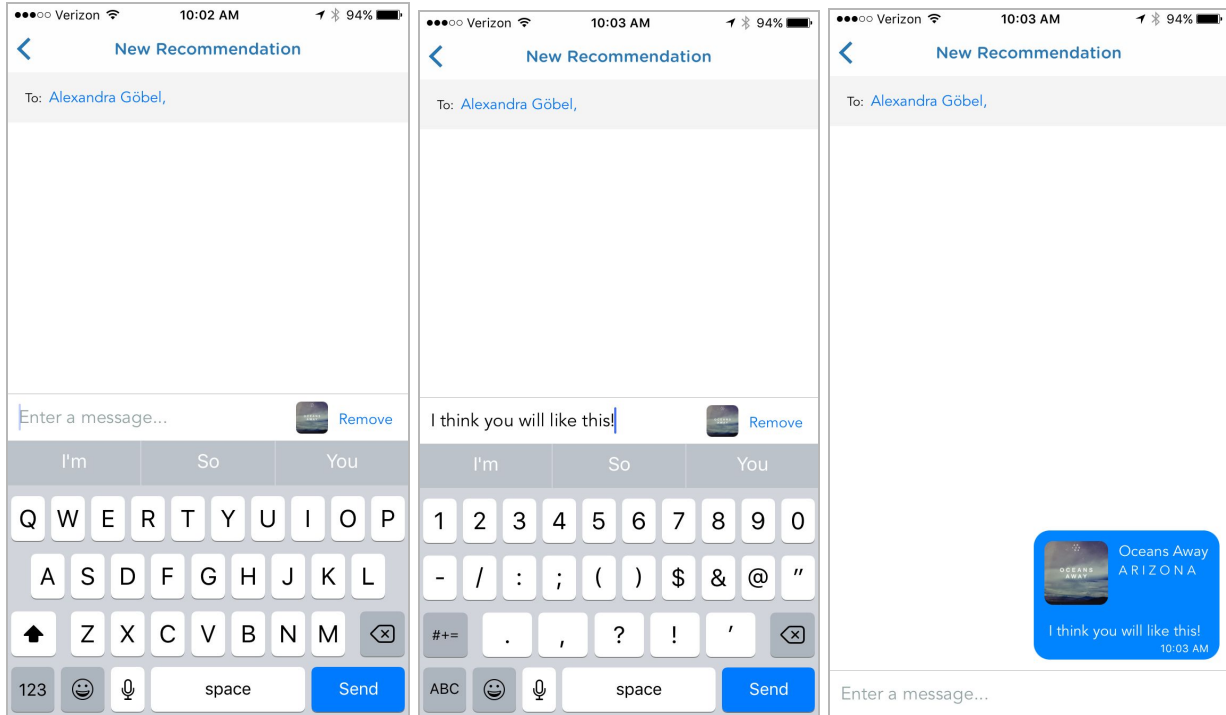


Figure 2: Once you have selected the song (left), type in a message to accompany the song (center). Click send and see the new recommendation appear in the conversation (right).

Task #2: Receiving a Song Recommendation (Moderate)

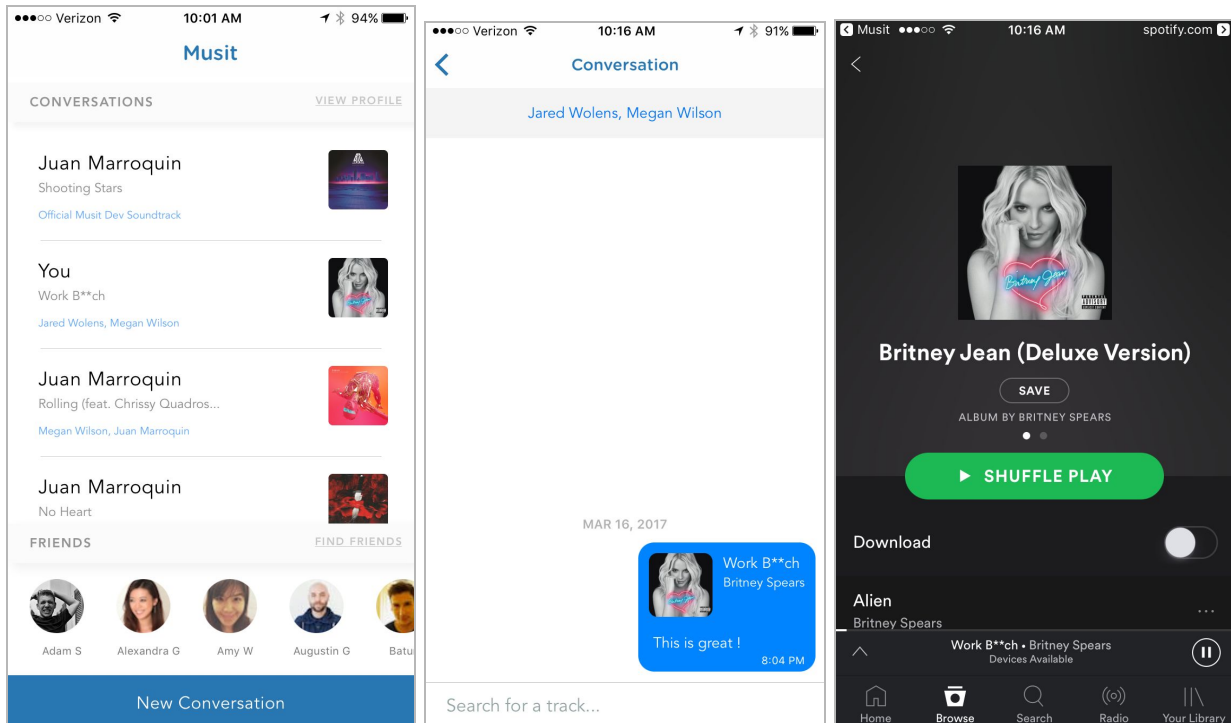


Figure 3: From the home screen (left), click on a conversation to view recommendations exchanged with this friend or group of friends. View the conversation (center), and click on the

album art in the recommendation to play the song in the service that the recommendation was sent for (right). Note that Musit uses deep linking to open song recommendations in Spotify and SoundCloud, and if the recipient does not use these platforms, the song will open in a web version of the service.

Task #3: Sharing Music with a Group

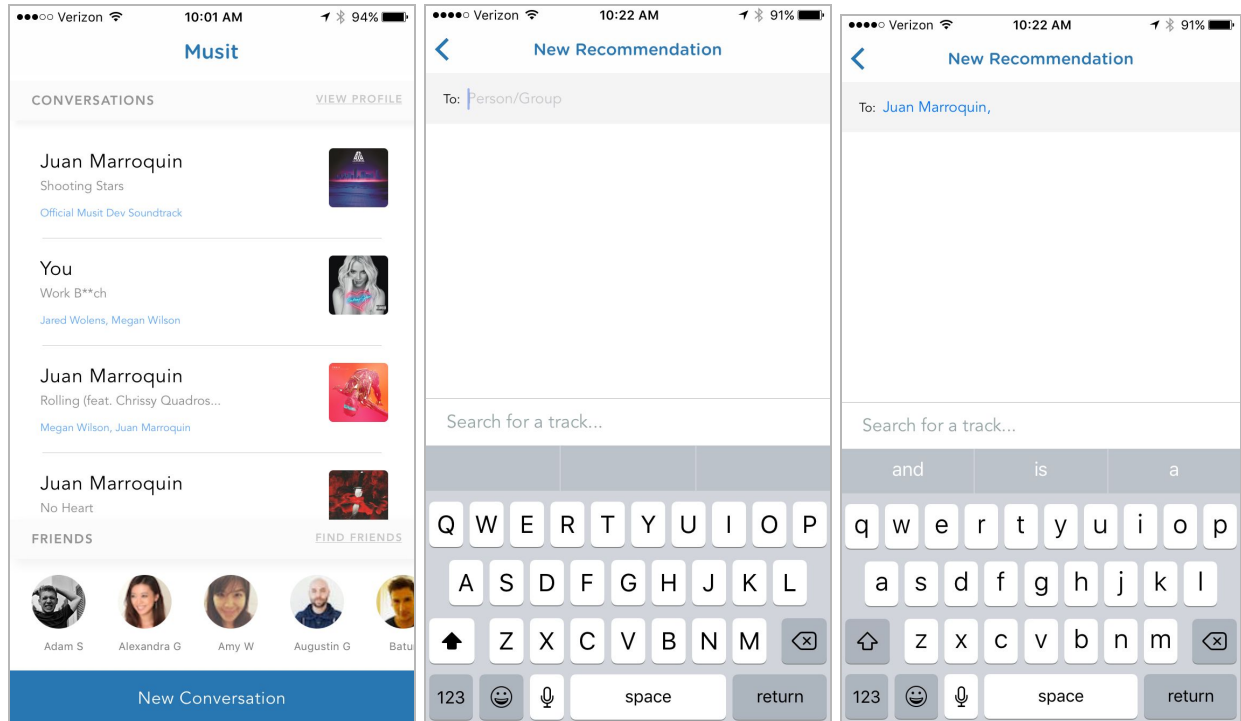


Figure 4: Open the app to the home screen (left) and click “new conversation.” Enter the names of the friends you would like to include in the group (center, right).

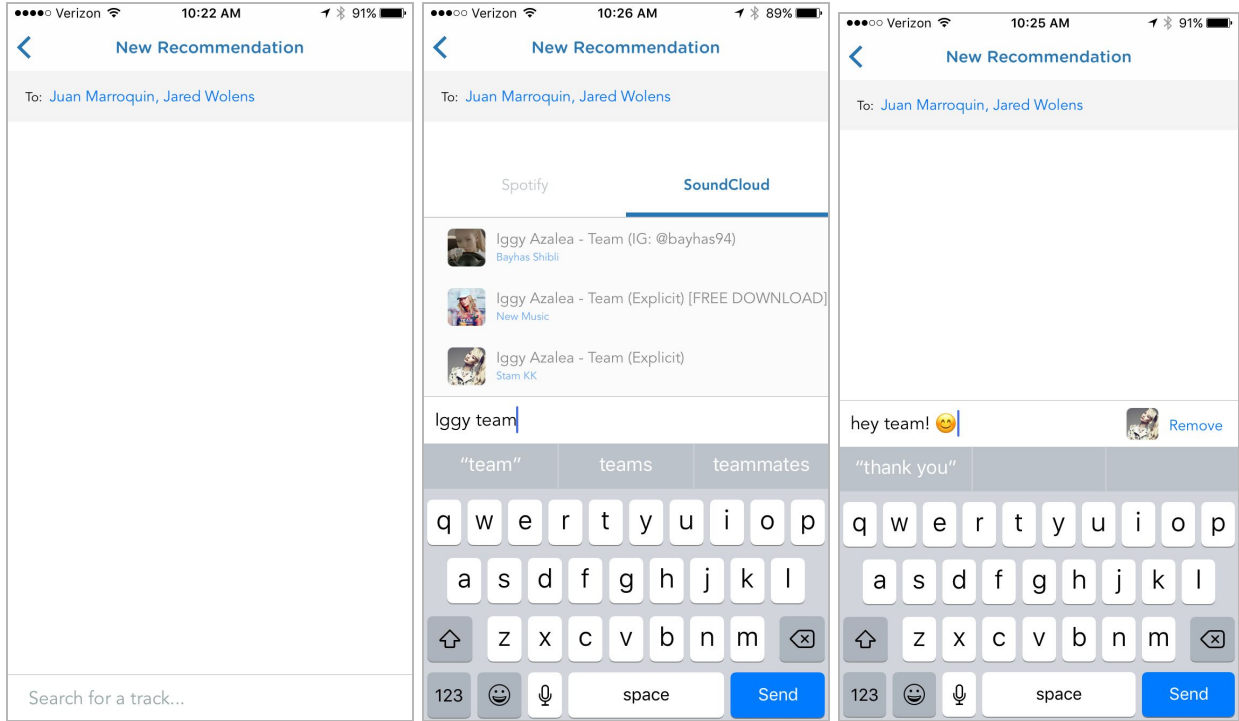


Figure 5: Finish entering the names of friends you would like to include in the group (left). Search for a song on SoundCloud or Spotify that you would like to recommend (center). Finally, type in a message to accompany the recommendation and click send (right).

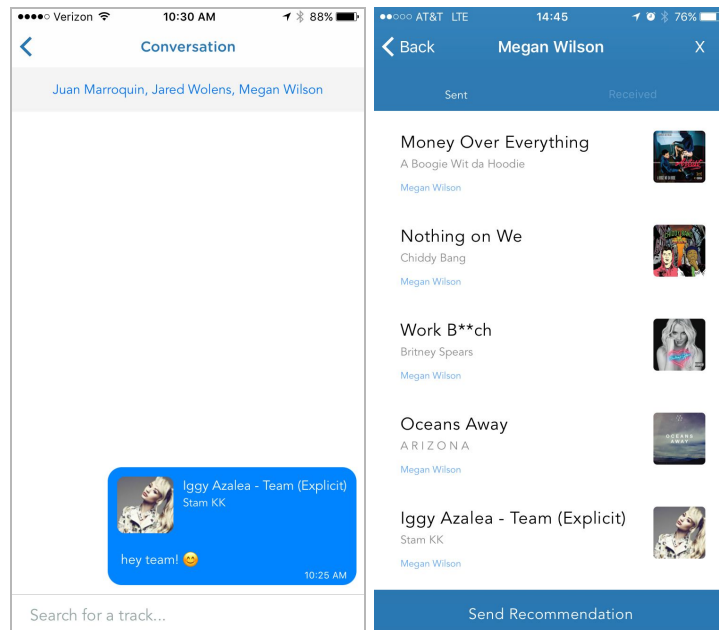


Figure 6: Group recommendation was sent and anyone in the group can click on the recommendation to listen to “Team” in SoundCloud (left). Another user can navigate to the poster’s page and discover their shared content (right).

DESIGN EVOLUTION

Design Iteration: Fall 2014

Lo-Fi Prototype:

The lo-fi prototype of our current app, “ShowMe”, was designed to help users send song recommendations, request recommendations from by constructing a playlist that others help populate, and explore trending music by location. The three tasks that this prototype helped users accomplish were: 1) discover popular music by genre and location, 2) submit a song suggestion to a friend’s playlist, and 3) create an empty playlist to solicit song suggestions.

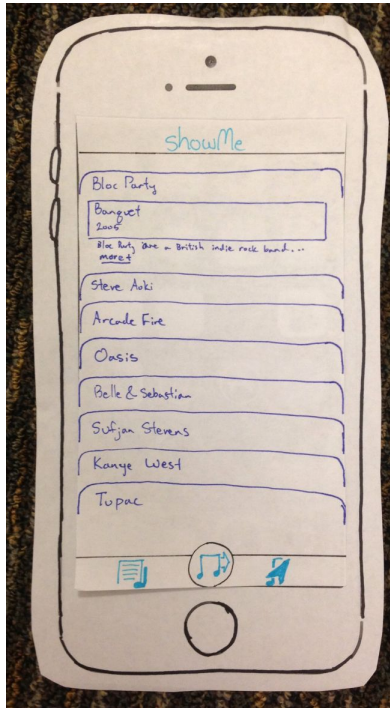


Figure 1.1: ShowMe home screen



Figure 1.2: Discover trending music by location and genre (task 1).

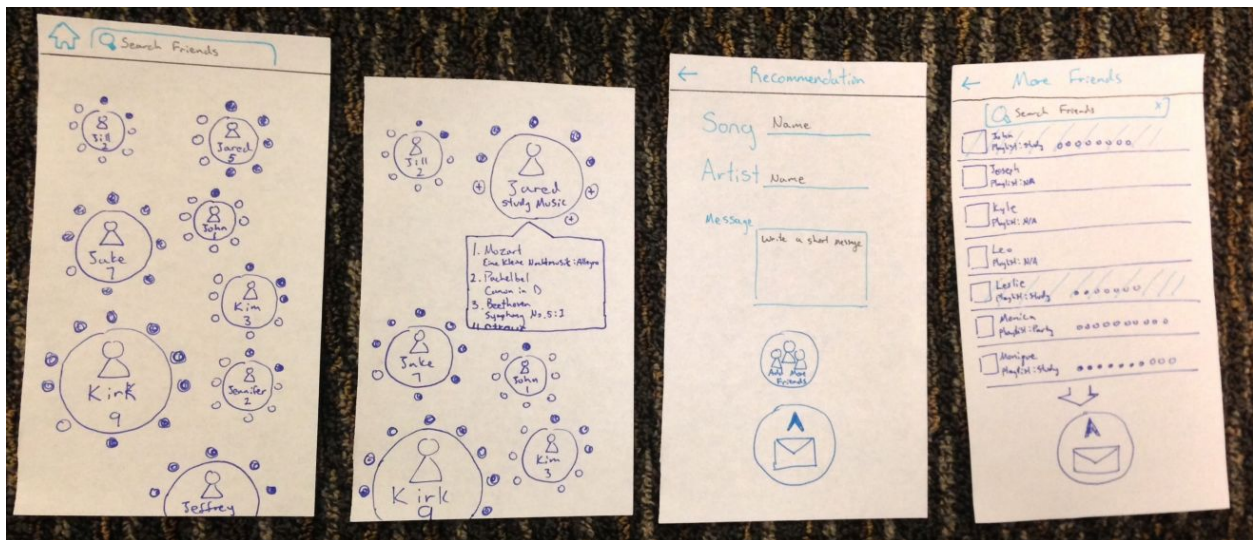


Figure 1.3: Send song recommendations to friends to help them populate their playlists (task 2).

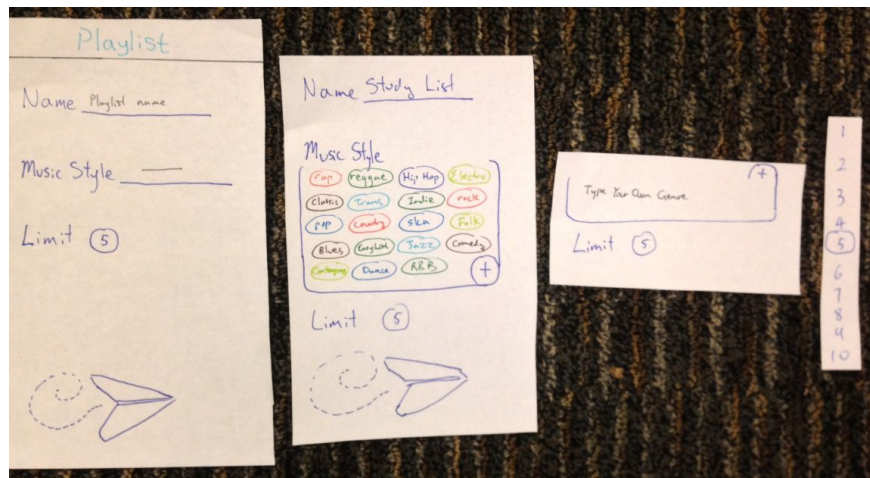


Figure 1.4: Create a new playlist and indicate playlist preferences. Use this playlist to solicit tailored recommendations from others (task 3).

Medium-Fi Prototype:

Several adjustments were made to ShowMe following user testing on the lo-fi prototype. The bulk of the feedback from usability study participants was that the design should be streamlined, and extra buttons and icons should be eliminated. Users also indicated that they would like to be able to create playlists with unlimited capacity, rather than having to limit their playlists to a certain number of songs upon creation.

The team refined the tasks as follows: 1) discover trending music by location and genre, 2) create playlists to curate song suggestions, and 3) share music suggestions to friends' playlists. They constructed the medium-fi prototype using Sketch and Invision. The task flows for each of the three tasks are shown below.

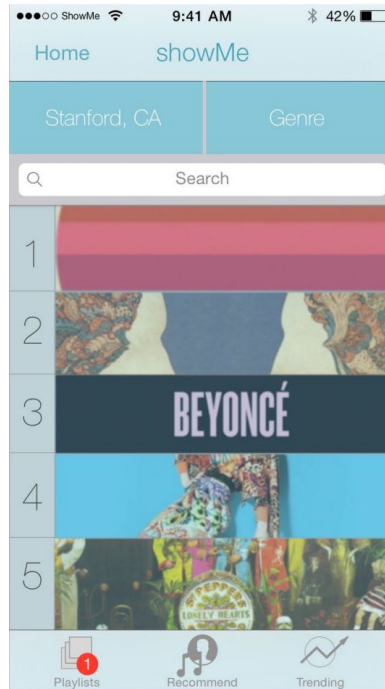


Figure 2.1: ShowMe medium-fi prototype landing page, where users can indicate location and genre for music discovery (task 1).

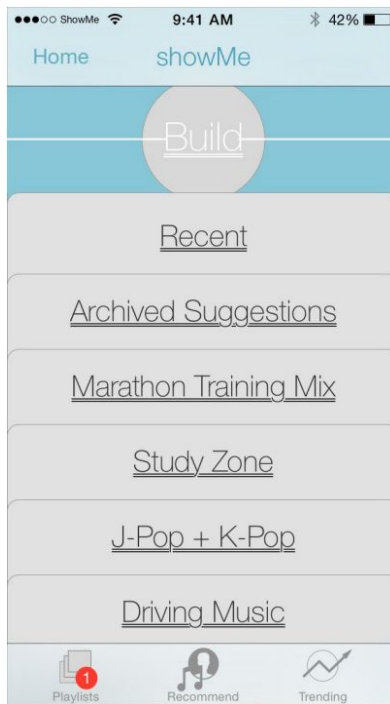


Figure 2.2: Create new playlist landing screen (task 2).

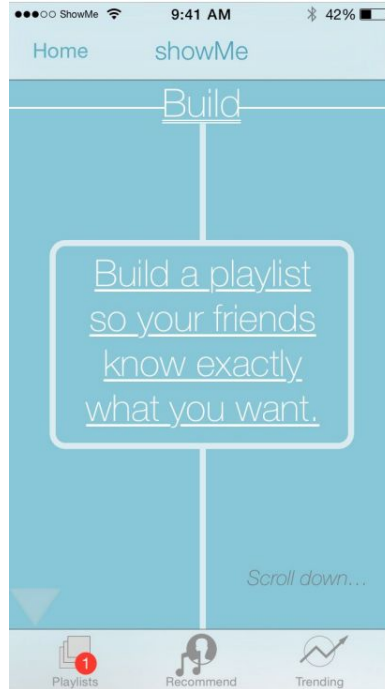


Figure 2.3: Begin creating a new playlist (task 2).

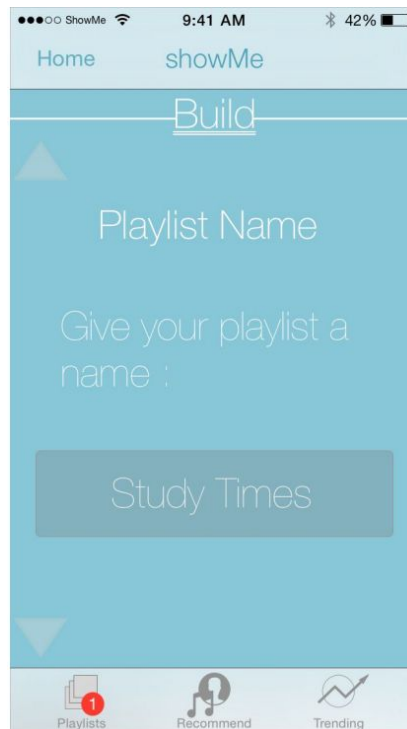


Figure 2.4: Indicate preferences when creating a playlist to give others direction when creating music recommendations (task 2).

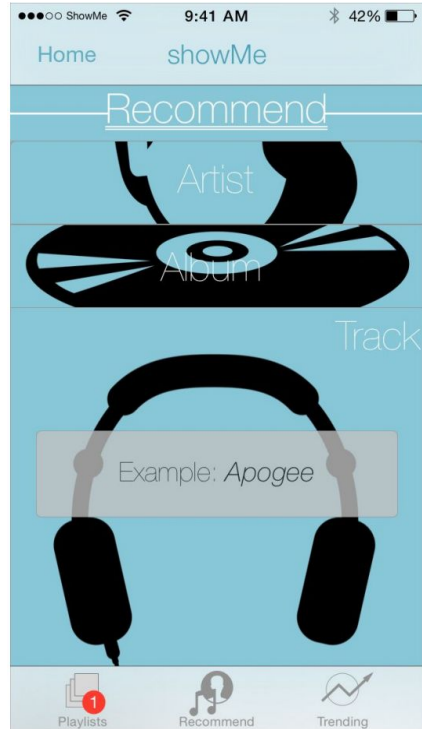


Figure 2.5: Send a recommendation screen, where a user can enter an artist, album, and track to send to friend(s) (task 3).

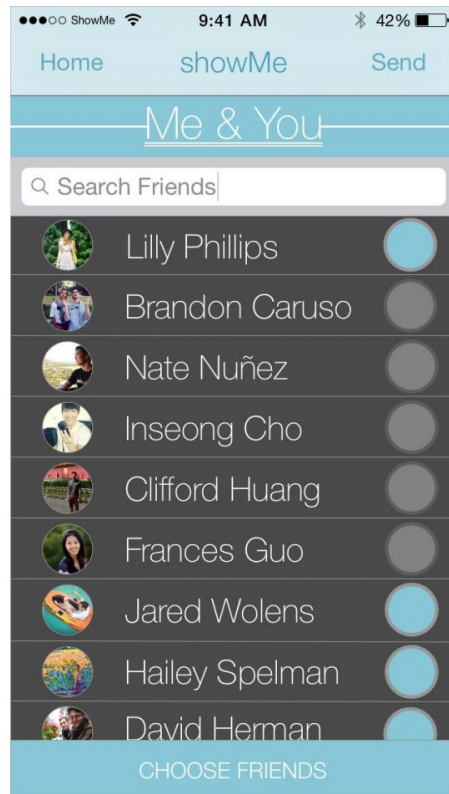


Figure 2.6: Indicate which friend(s) to send the recommendation to (task 3).

The task flow for discovering music by location and genre did not change from the lo-fi prototype to the medium-fi prototype, however the UI was streamlined to eliminate clutter and make entering preferences clearer. The second task, creating playlists to curate song suggestions, was changed to allow users to create several different playlists to curate suggestions, which contrasts the previous design iteration where they were only permitted to create one playlist to receive song suggestions. The third task, sending a song suggestion to a friend, was also expanded to allow users to easily select several friends to send a recommendation to, rather than limiting each recommendation to a single friend.

Hi-Fi Prototype:

After conducting a heuristic evaluation on the medium-fi prototype, the team synthesized user preference data and constructed a hi-fi prototype using Swift and XCode. The three core tasks that the app facilitates remained the same from the last iteration.

The team redesigned the home screen and the discovering trending music by location and genre flow by making the design more minimalist. Reducing the number of images on the page and switching from a blue background to a white one streamlined the discovery process and made the differentiation between tappable and non-tappable regions more clear.

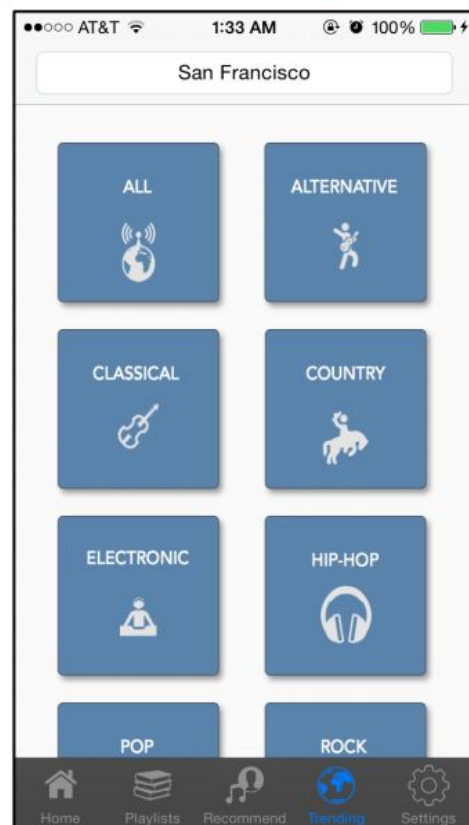


Figure 3.1: Discover trending music by location and genre landing page and Musit home screen (task 1).

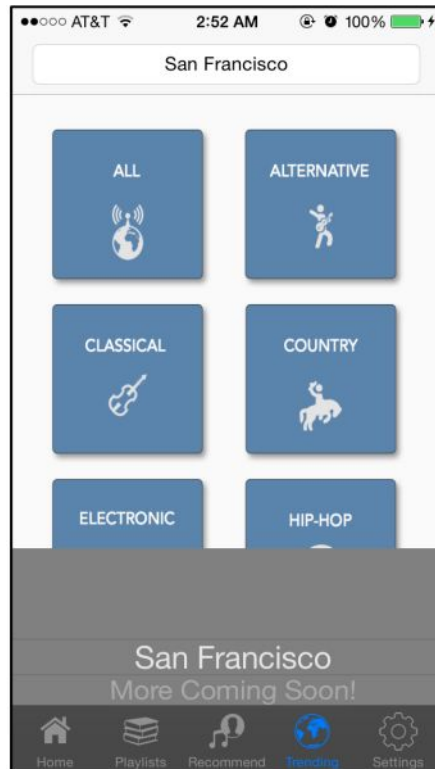


Figure 3.2: Use the search bar at the top to find a location, and click on genre buttons to find trending music for each of these genres in the indicated location (task 1).

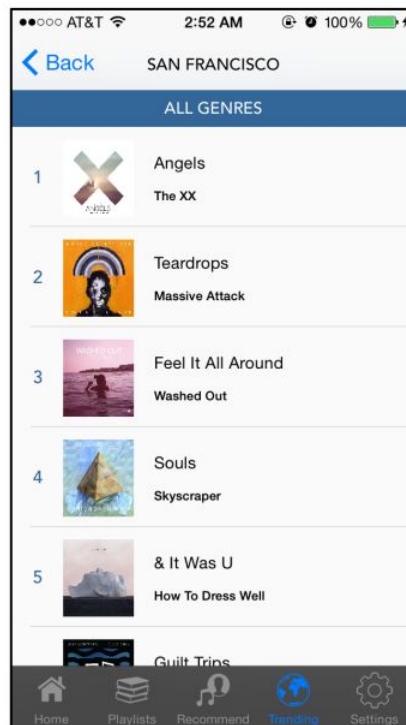


Figure 3.3: View popular music in all genres in San Francisco (task 1 example).

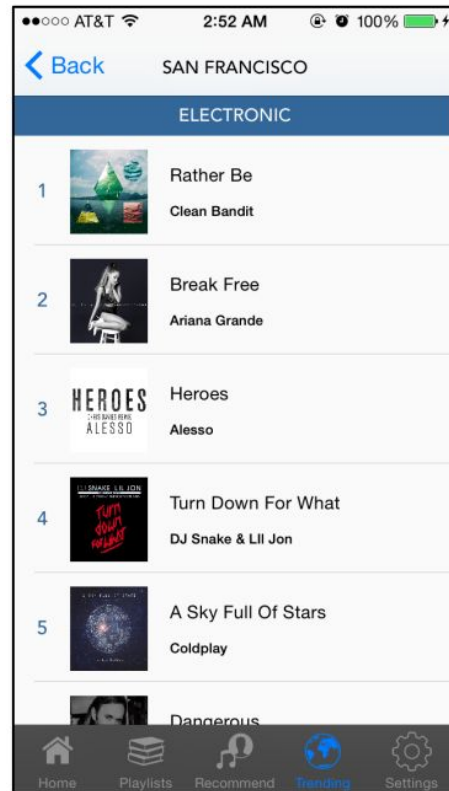


Figure 3.4: View popular electronic music in San Francisco (task 1 example).

The flow for creating a playlist in the medium-fi prototype requires users to scroll through several pages to indicate their preferences for their playlist. For example, after they indicate the name of the playlist, they must scroll to a new page to enter the situation they would like to listen to music in and other details about the playlist. Users found this scrolling through pages to indicate preferences cumbersome, and, as a result the hi-fi prototype uses only one screen for creating a playlist.

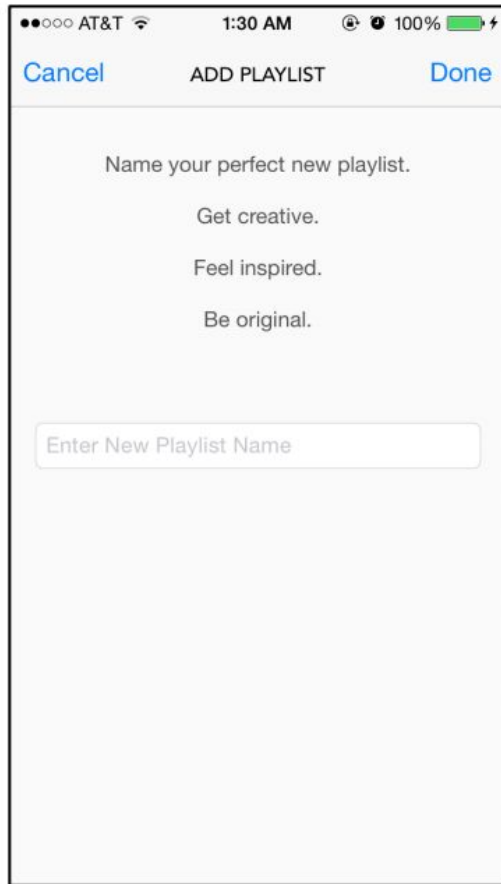


Figure 3.5: Create a new playlist page (task 2).

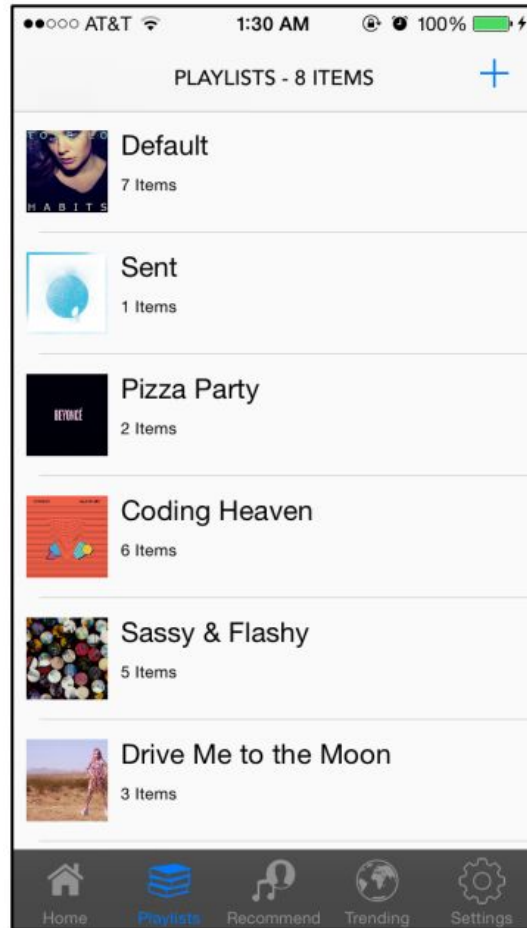


Figure 3.6: Playlists landing page, where a user will be taken when they navigate to “Playlists” using the tab bar, or immediately after creating a new playlist (task 2).

The design of the sending a song recommendation task was streamlined in a similar way to the design of creating a new playlist. In this prototype, the team removed distracting background graphics and replaced all input areas with traditional input fields to make the intended user interaction clear.

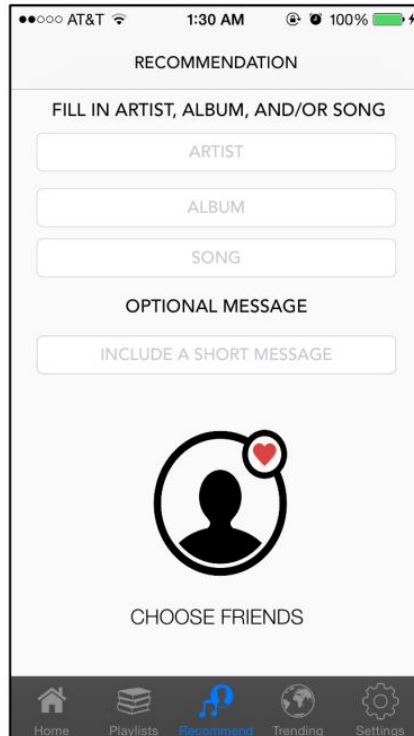


Figure 3.7: Simple and straightforward input for creating a new recommendation (task 3).

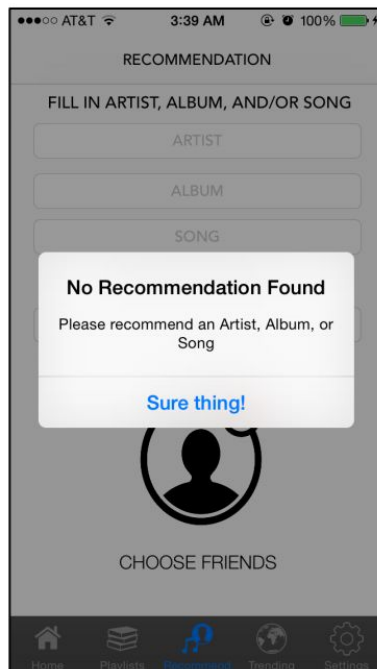


Figure 3.8: Error checking to ensure that recommendation input fields are populated and formatted correctly (task 3).

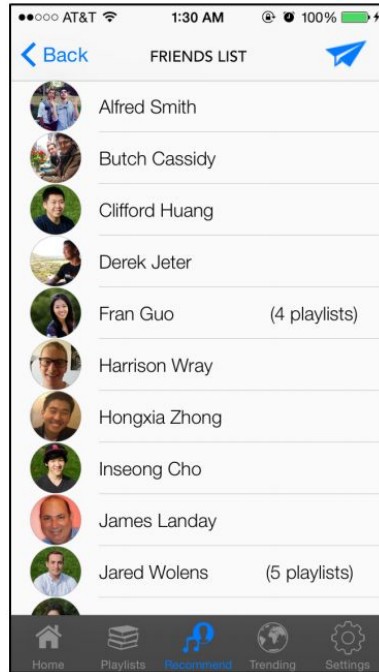


Figure 3.9: Select which friend(s) to send the recommendation to (task 3).

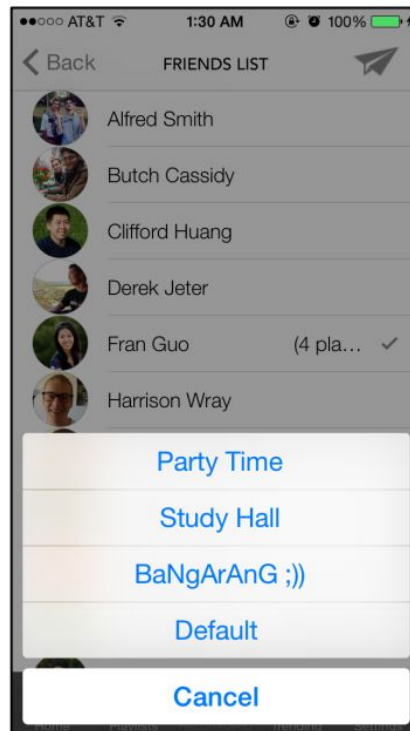


Figure 3.10: If a selected friend has multiple playlists, select which playlist to share the song to (task 3).

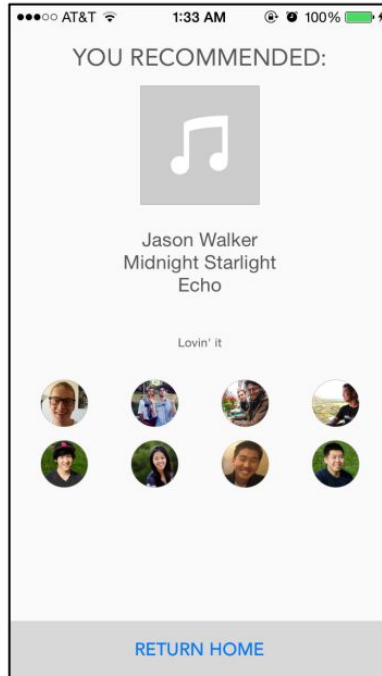


Figure 3.11: Confirmation screen that indicates the details of your recommendation as well as the friend(s) the recommendation was sent to (task 3).

Overall, the functionality from the medium-fi prototype to the hi-fi prototype stayed mostly the same. The principle changes in this iteration were simplifying the design and decluttering the interface so users could work through each of the core tasks more seamlessly.

Design Iteration: Winter 2017

Hi-Fi Prototype #1:

At the beginning of the quarter, our CS 194H team made significant updates from the hi-fi prototype that the CS 147 team left off with. First, we switched to using Angular and React Native to re-implement Musit as a web app. We also implemented the Echo Nest and Spotify APIs for querying and auto-search completion. For the back end, we removed the hard-coded data for three data collections: playlists, recommendations, and users. Finally, we adjusted the front-end design to comply with web app standards rather than Android.

Overall, the concept of the app for this iteration was largely the same as the concept that the CS 147 team left off with. This prototype aimed to facilitate the completion of the following three tasks: 1) discover trending music by location and genre, 2) send music recommendations to friends' playlists, and 3) solicit music recommendations from others by creating a playlist. Images of screens for completing these tasks are shown below.

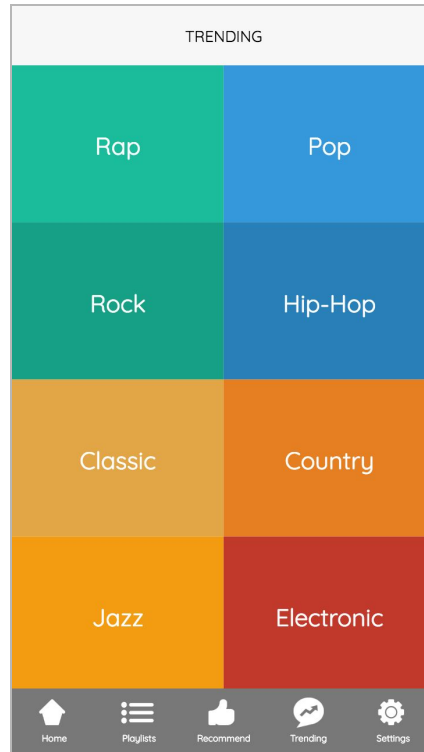


Figure 1: When the trending tab is clicked in the tab bar at the bottom of the screen, the user can choose among several different genres to see what music is trending in their area for the selected genre (task 1).



Figure 2: When the recommendation icon is clicked in the tab bar, the user is taken to a blank recommendation form to be filled out. Here, they can search Spotify for a track, artist or album to recommend. Selecting a song populates the “song title” and “artist” fields, and adds album art in the placeholder spot. The user can then add a message to the recommendation and choose friend(s) to send this recommendation to (task 2).

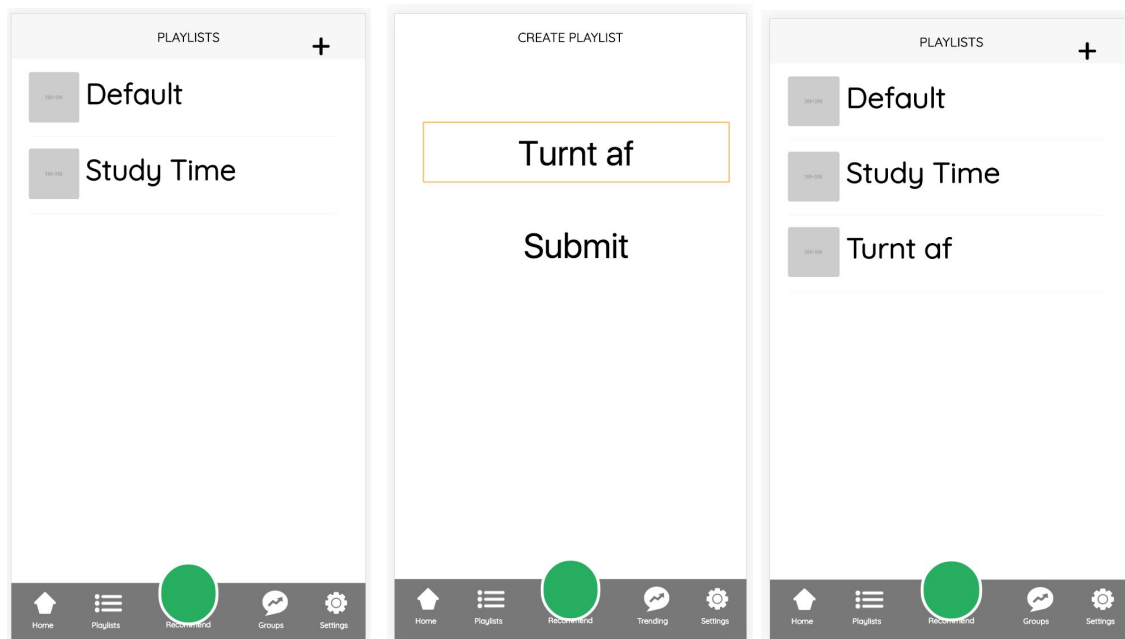


Figure 3: This task displays a trialed recommendation button. From the playlist screen, users can press the ‘+’ icon to add a new playlist, at which point they simply tag the playlist with name and hit submit

Hi-Fi Prototype #2:

Our team conducted usability study testing on our web app prototype and received feedback that led to several major conceptual and design changes. First, we learned that the trending feature (task #1 in our last prototype) does not add value to our application or help us differentiate from other existing music services. Many other music services including Spotify, Pandora, and SoundCloud use algorithms to help users discover music, and adding this functionality to our app does not help us accomplish our goal of facilitating music-based relationships. Second, we learned that several aspects of our send a music recommendation flow were unintuitive and clunky. Users wanted to be able to do bulk sends of a single song rather than having to recreate a recommendation several times to send a song to multiple friends’ playlists. We also learned that we should encourage collaboration on the app by restructuring the permissions for playlists, so they can be as public or private or as collaborative or individualized as the user decides. Finally, we learned that people are nervous about soliciting suggestions from others and that they need constant interaction and emotional communication to stay engaged. We used this finding to flesh out a home feed that keeps in tune with their friends’ activity and needs, as well as a schema for communication on Musit.

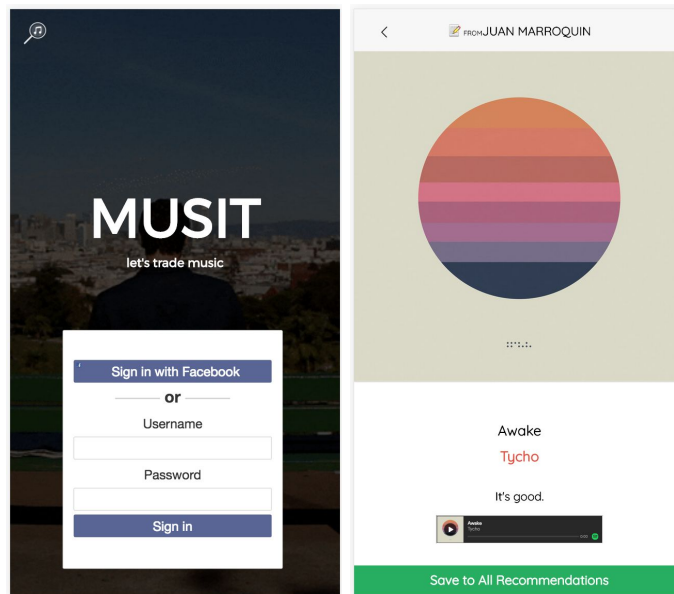


Figure 1: User accounts were created with Facebook to allow users to access their account anywhere (left). Individual recommendations are accompanied with recommendation screen. From this page, users can see all the metadata about a song, as well as listen to a snippet of the track from the app. We offered users the option to save a recommendation only if they enjoyed the track (right)

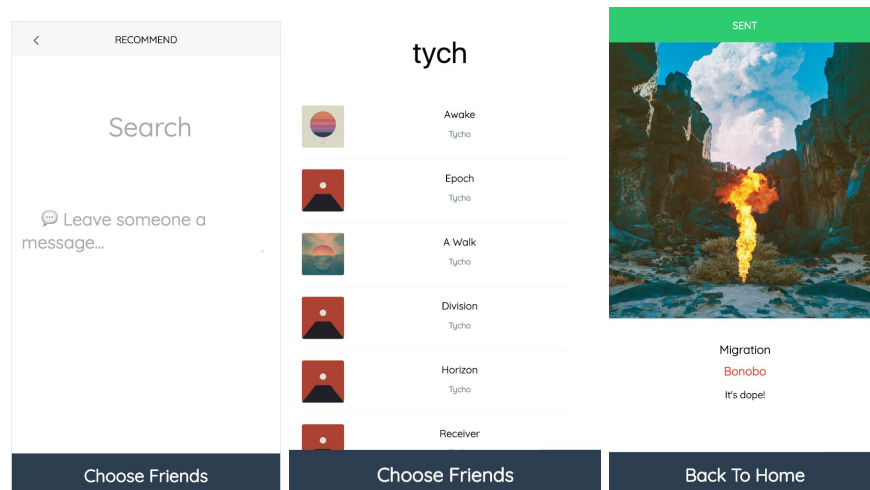


Figure 2: Simplified recommendation screen using fewer form fields (left). We implemented a more rigorous search to display greater feedback about the songs individuals are choosing. The songs are collected via queries to a public Spotify API (center). Final screen demonstrates the confirmation of sending page for a user (right)

Hi-Fi Prototype #3:

The Musit team developed two iterations for this part, both centered around the idea of a “music messenger”. We rapidly build our first iteration and conducted a lab usability study using this prototype. This lab usability study guided us in making updates for our final hi-fi prototype #3. In this usability study, we learned that the transition from a traditional messaging app to a messaging app where music is exchanged is intuitive, but there are a few areas where extra guidance would facilitate the transition. We also learned that music sharing is just as much about the people as it is about the content being shared, and that people want to turn recommendations they receive into recommendations they share with others. Keeping these findings in mind, we made several adjustments for our final prototype.

The first adjustment we made was to update the home screen. We updated the previous three categories (“Recent Recommendations,” “Threads,” and “Musit Network”) to two categories (“Conversations” and “Friends”). We made this change after user feedback which indicated that it would be most intuitive to keep group and individual recommendations in a single inbox category, as well as the feedback that users would like to be able to view profiles of individual friends.

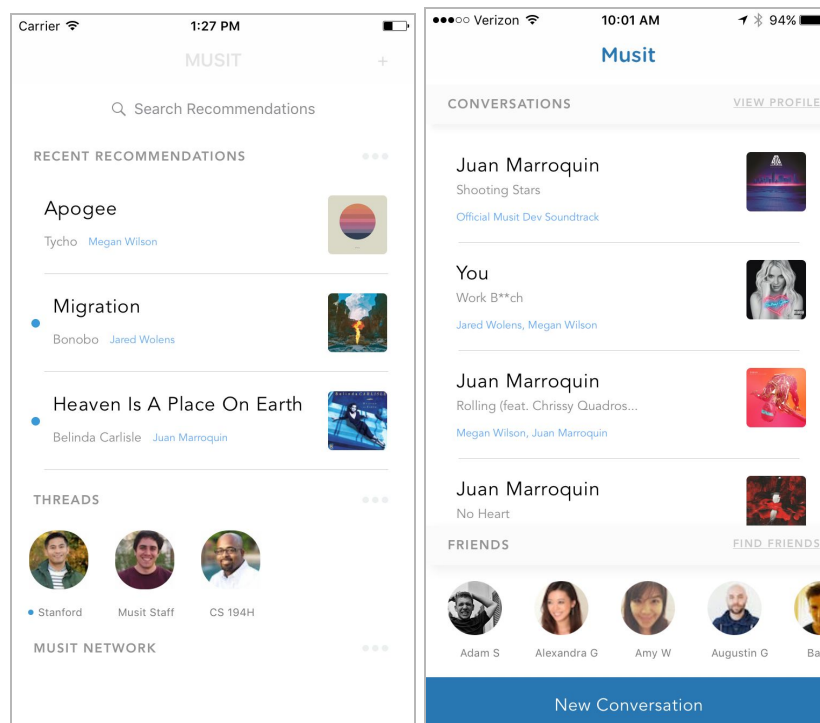


Figure 1: Previous home screen with “Recent Recommendations,” “Threads,” and “Musit Network” categories (left) and new home screen with “Conversations” and “Friends” categories.

We then updated the home screen to make it clearer how to begin the task of sending a recommendation. To do so, we replaced the small, almost invisible “+” icon in the top right corner of the interface with a large “new conversation” button at the bottom of the screen.

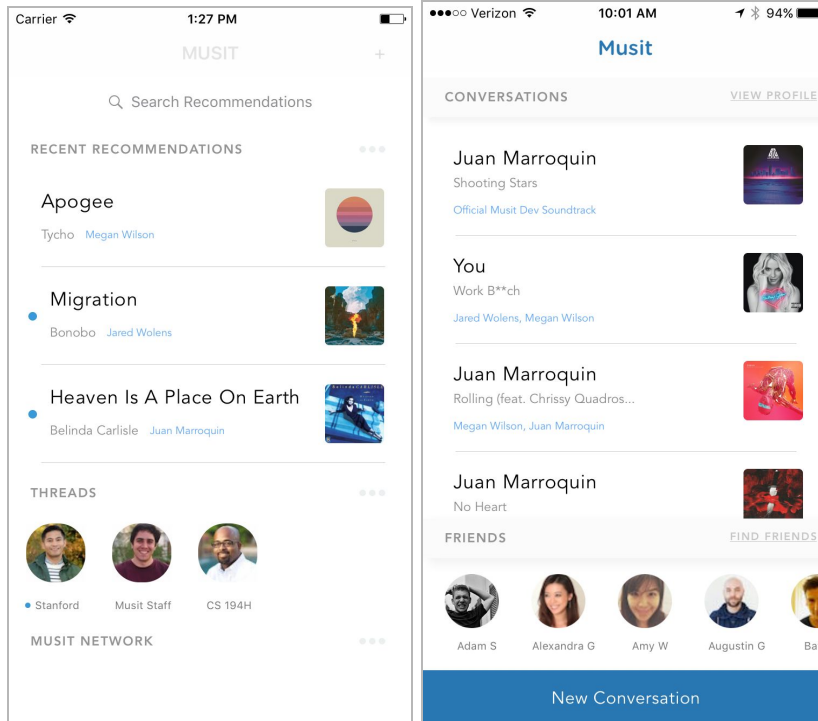


Figure 2: Icon in top right corner of previous home screen is nearly invisible (left), whereas the “New Conversation” button on the new home screen is visible and actionable (right).

We also adjusted the guide text in the input box for sending an individual recommendation to indicate that users should type in the name of a track on Spotify or SoundCloud to eliminate formatting mistakes when trying to send a recommendation.

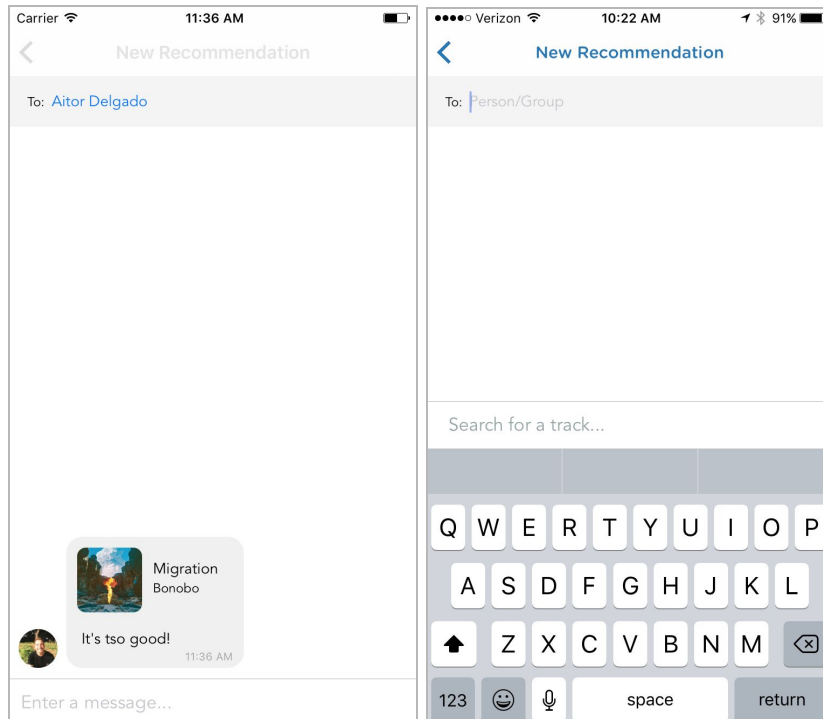


Figure 3: In the previous design (right), the guide text for entering a song recommendation, “Enter a message...”, left users confused about how to format their recommendation. The guide text in the updated design (right), “Search for a track,” makes it clear that the user should enter a song name.

Next, we decided to integrate SoundCloud recommendations to broaden the reach of our app, as user feedback suggested that adding SoundCloud integration would open our app to more music “power users.”

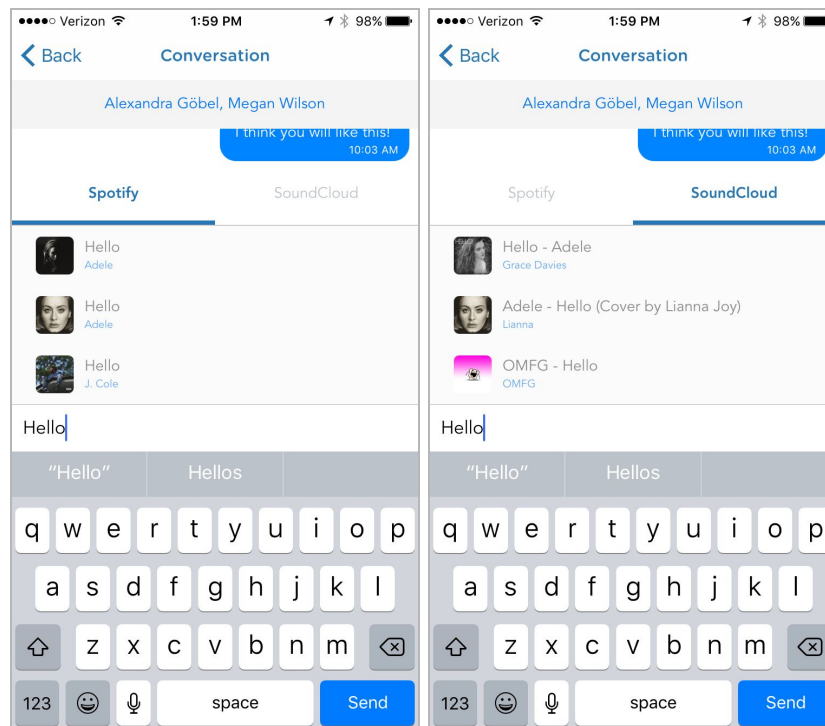


Figure 4: Hi-fi prototype #3 update allows users to choose to send their recommendation from Spotify (left) or SoundCloud (right).

Finally, we updated the way that users view the recommendations they receive based on the feedback that the people involved in music sharing are just as important, if not more important, than the content being shared. To address this point, we restructured the way a recommendation is displayed in the inbox view. The names of the individuals involved in the conversation (or the group name in the case of group recommendations), is displayed on its own line to emphasize the people involved in the exchange.

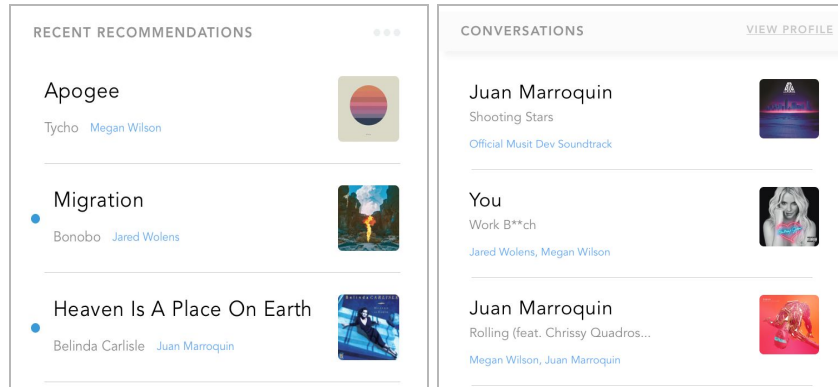


Figure 5: In the previous prototype (left), the names of those involved in the conversation were displayed on the same line as the artist name of the content being recommended. In our hi-fi prototype #3, the names of those involved in the conversation are displayed on a separate line from the song information to emphasize the people in the sharing relationship (right).

FINAL INTERFACE

Final UI Design

Musit helps users complete three fundamental music-sharing tasks: sending song recommendations to individuals, receiving song recommendations from others and interacting with these recommendations, and sending and receiving music recommendations as part of a group. At its core, Musit is a music messenger that facilitates the development and maintenance of music-based relationships.

The final design of Musit features an inbox home view that allows users to keep track of all of their music recommendations in a centralized location. This design decision directly solves the problem of music recommendations getting lost in old text message threads. From this home view, users have the ability to complete each of the core tasks.

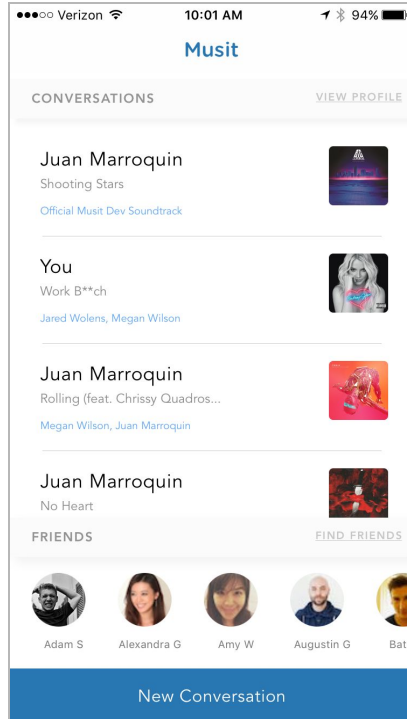


Figure 1: Musit home screen featuring inbox view of recent recommendation exchanges and friends.

To send a recommendation, the user can click on the “new conversation” button at the bottom of the screen to pull up a blank recommendation. From there they can type in the name of the friend(s) they would like to send the recommendation to, the song they would like to send, and an accompanying message if desired. Alternatively, a user can author a new recommendation by clicking on the icon of a particular friend in the “Friends” section of the home screen or by clicking on a recent conversation in the “Conversations” section of the home screen. In either case, a conversation will be pulled up and the user can type in the input box at the bottom of the screen to send a new recommendation to continue this conversation.

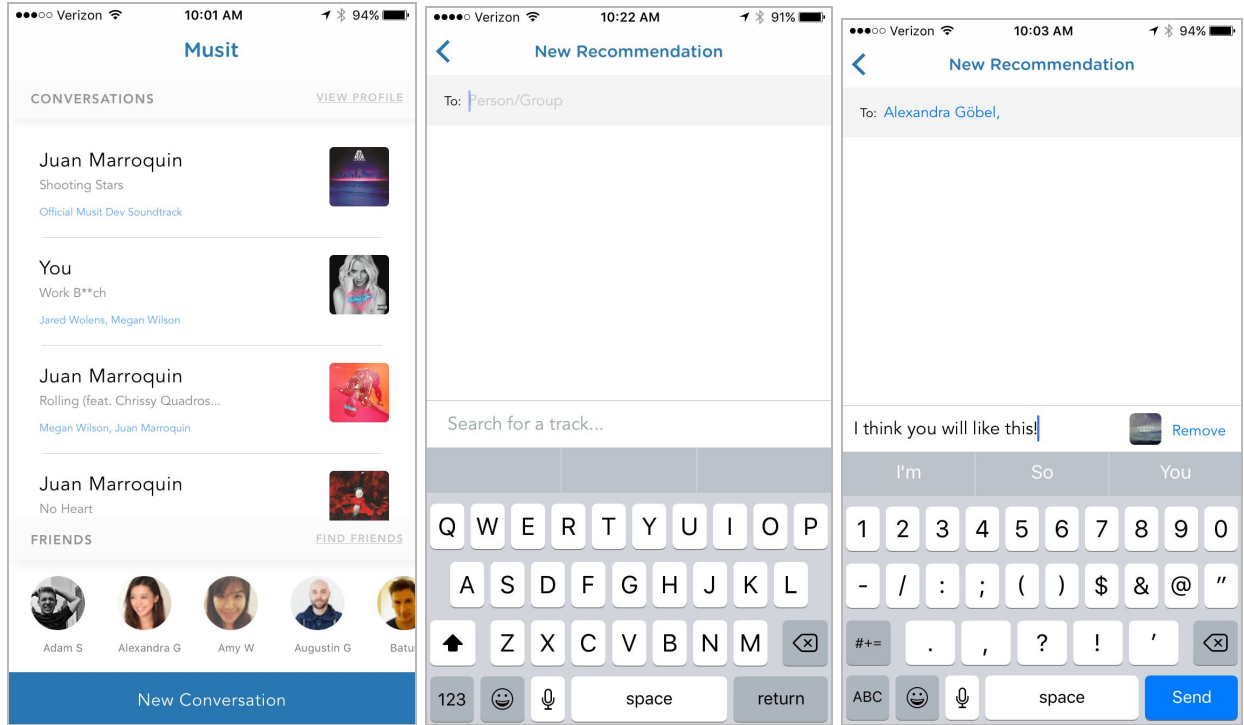


Figure 2: Send a recommendation using the “New Conversation” button.

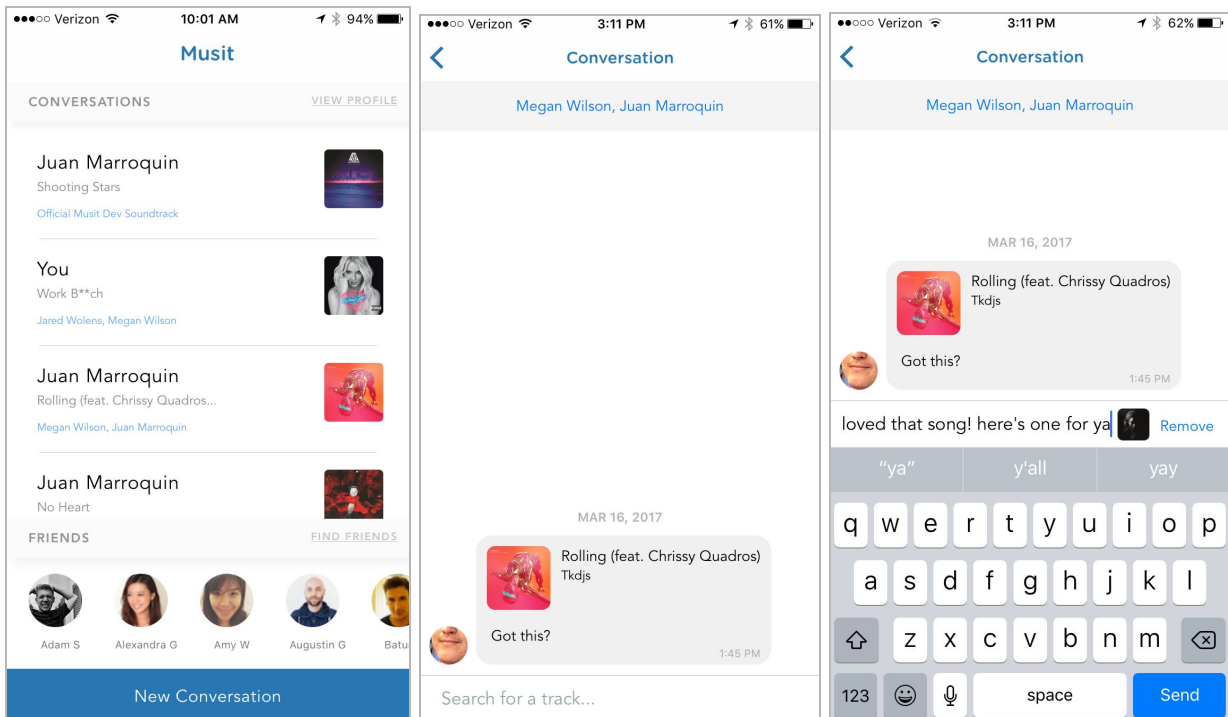


Figure 3: Open an existing conversation to send a new recommendation.

When a user receives a new recommendation, they will get a push notification from Musit. Upon opening the app to the home screen, they will see this new recommendation at the top section of the “Conversation” sections, as this section of the inbox view contains recommendations in the order they were sent/received. The user can click on the conversation and tap the album art icon to listen to the song in Spotify or SoundCloud, depending on which service the recommender sent the song in.

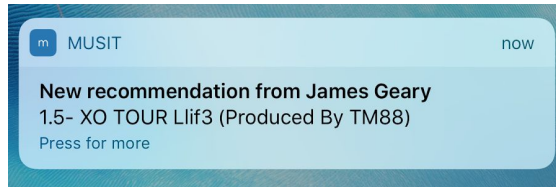


Figure 4: Musit push notification.

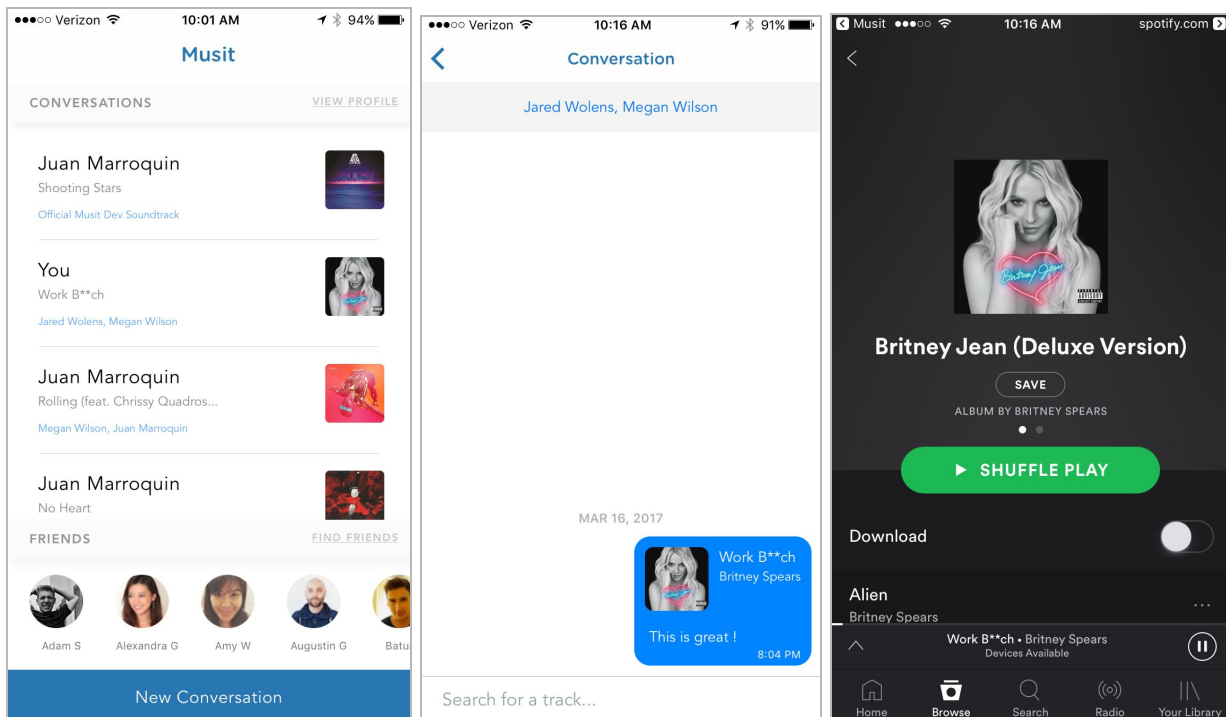


Figure 5: Receive a recommendation and listen to the recommendation on Spotify.

Interacting with groups on Musit is very similar to interacting with individuals. Conversations with groups will appear in the same format as a conversation between two individuals in the “Conversations” section of the home screen, with the exception that the names of all members of the group will be displayed rather than just the names of the two individuals involved in a one-on-one conversation. A user can click on a group conversation to view the past interactions the group has had, and to send a new recommendation to the group. Musit users can create their own group any time by clicking the “New Conversation” button at the

bottom of the home view, typing in the names of two or more friends, and sending a recommendation using the same flow as they would to send a recommendation to an individual.

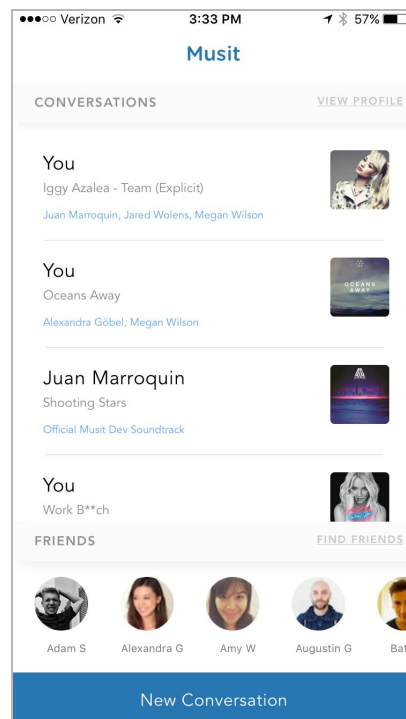


Figure 6: The “Conversations” section of the home screen contains recommendations from between two individuals as well as group recommendation conversations. Conversations between two individuals have the two participants names listed in blue, whereas recommendations among groups list the group name if one has been established, or the names of the participants of the group if no group name has been established. For example, the group of Megan, Jard and Juan lists all three names, whereas the recommendation Juan sent of Shooting Stars was sent to the “Official Musit Dev Soundtrack” group.

Features Left Unimplemented

As we approached the end of the quarter there were a number of features we wanted to include but did not have the time for. Some of these were meant to make the app more convenient, while others were meant to create richer conversations, however we currently have all of the features necessary to release the app to the public and have it perform its core functions.

One feature we wanted, but did not have time for, would allow us to create Spotify playlists on the user's behalf using the songs in a conversation. We felt that this would make it more convenient for a user to listen to multiple songs in a row, however implementing this would have involved adding another native SDK to our project. We decided instead to go broad with our music services and add SoundCloud to allow creators and fans of smaller musicians to share songs.

We also wanted to add the ability to clearly acknowledge a user's recommendation. We found that conversations can devolve into two people sharing recommendations blindly without actually responding to the other person, and a visual indication of direct response may be useful in remedying this. This may have taken the form of either threads, in which users could respond to recommendations in a nested conversation, or something similar to iMessage's reactions where a user could simply "thumbs up" a recommendation or react in some other way.

Aside from these two features, there were other more straight-forward improvements such as deep-linking notifications and adding more music services such as Apple Music and YouTube.

Tools

For our final product we used React Native to create a native iOS app. React Native worked well for us since our team was much more familiar with web development than native development which made the transition to React Native from our previous Angular prototype much easier. One caveat to using React Native was that some of the features we wanted to use, such as native Spotify and Facebook support required integrating native SDK, which we were not as comfortable doing.

Our database was Firebase, which was a first for all of us. Firebase was useful because it's API allowed us to very easily implement real-time updates for things like the messenger component and the home screen. However, Firebase did take some getting used to as we transitioned from MongoDB, since Firebase does not have the ability to make complex queries, in fact basically we were limited to querying by id. Figuring out how to structure our data and our calls and how to cache data was an important, and satisfying step in making Firebase work for us.

For our notifications, we built a simple Node server that used the same Firebase SDK to detect when a user sends a message and to dispatch notifications. This was fairly straightforward since we were able to reuse much of the same code that we had already built for the base app.

For music retrieval we used the Spotify and SoundCloud APIs. These two APIs were very easy to use, with the slight issue that the data they returned was not formatted the same way. When we added SoundCloud to our app, we ended up taking the data returned by the SoundCloud API and remapping it to look like the Spotify API so that we could effectively treat them the same.

Downloading Musit

To download Musit on the App Store, visit bit.ly/musit , or search for "Musit" on the App Store.

MAKING IT REAL

Our Team:



Juan Marroquin, B.S. in Computer Science '17

Juan's background is in web development and UI/UX. He is passionate about making interfaces that can work on today's cumbersomely large screen sizes. His role on the Musit team focused on full-stack development and usability.



Jared Wolens, M.S. in Computer Science '17

Jared has experience in UI/UX, design, and web development. He has a passion for building products surrounding community outreach and public art. His role on the Musit team focused on front-end development and design.



James Geary, B.S. in Computer Science '17

James has experience in building web applications and product management. He has recently been very interested in blockchain and Ethereum. His main role on the Musit team was to set up and manage the constantly evolving stack; in addition, he worked on development and design.



Megan Wilson, B.S. in Computer Science '18

Megan has experience in UI/UX design, product management, and front-end web development. She enjoys working on projects that bring people from diverse backgrounds together and is passionate about building products that help users inject creativity into their everyday lives. Her role on the Musit team focused on usability and design.

Why we are qualified:

Given our collective experience in web development, we are well qualified to take advantage of existing music services' APIs. Additionally, we can take advantage of our own experience with sending music on existing platforms as well as our UX background to identify pain points in the process of sending music. This allows us to design and build a better music recommendation process.

Business Model:

Current music platforms have allocated sharing as an auxiliary service. Services such as Spotify rely on unconscious metrics such as listening habits to discern potential interests of users and suggest them new artists and concerts based on complex learning algorithms. Musit has the potential to understand entire groups of people's habits and interests and suggest them concerts/artists in real-time based on direct conscious habits of the user. Collaboration with production studios could result in strategies ripe for both monetization of advertising concerts to users as well as acquisition by larger platforms such as Spotify and Pandora to encourage more habit-forming behaviors. Spotify currently faces the challenge of procuring music at a rate that matches the habits of their users; however, more than 4 million tracks (20% of all Spotify songs) have gone unlistened to. Scraping user conversations allows Musit to sell back to streaming services information pertaining to trending conversations to better direct streaming services actions of track procurement and contract forming.

Usage of music platforms is fairly ubiquitous with roughly 68% of US-based smartphone users listening to music via a streaming service, with Spotify and Soundcloud both occupying spots in the top 3 services. Market research suggests our most active group for sharing music is 18-35 year olds with an equal split in demographics of users suggesting our market size surpasses upwards of 60 million Americans when solely considering US-market. At Musit, we aim to change the current model of music sharing to be one that fosters friendship, ease, and ultimately, a love of music.

SUMMARY

Musit streamlines the music sharing process and helps individuals keep track of the content they send to and receive from their friends. Making music sharing more convenient encourages individuals to engage in music sharing more frequently, and, as a result, will help people deepen their relationships with those around them. Our team believes that Musit has the ability to make connections between individuals and groups more profound by facilitating frequent engagement with others around creative and expressive content.