

Jorge Pozas Trevino, Robert Fearon, Emmerich Anklam
4 December 2014
Interactive Prototype Report

soneme

encouraging the cooperative discovery of lesser known artists

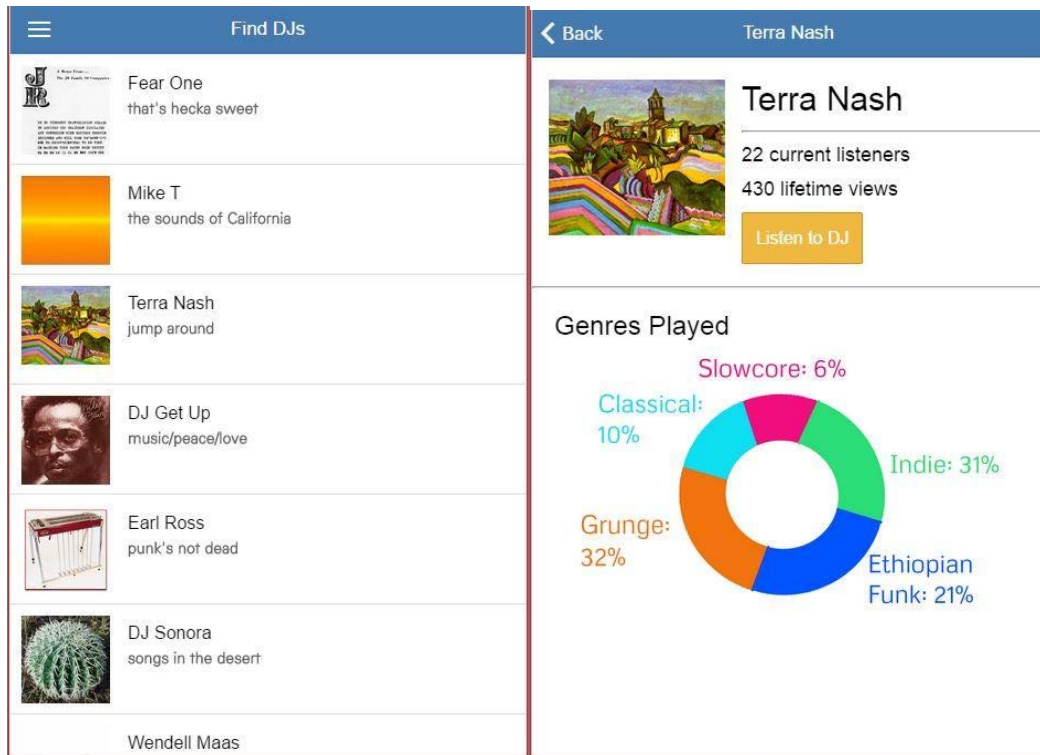
Problem and Solution Overview

Even with the current proliferation of tools for hearing music, there is a major lack of new applications directed specifically toward cooperative music discovery. Soneme introduces an interface that helps its users find and share great, lesser known music in just a few minutes. The app aims for an exciting listening experience that prioritizes finding music over archiving users' favorite songs. It also seeks to make music discovery a vivid social experience, letting everyone broadcast their discoveries in real time, in the vein of classic radio DJing.

Tasks & Final Interface Scenarios

Even with our design changes from our medium-fi to our hi-fi prototype, our tasks are still largely consistent with those for our medium-fi prototype.

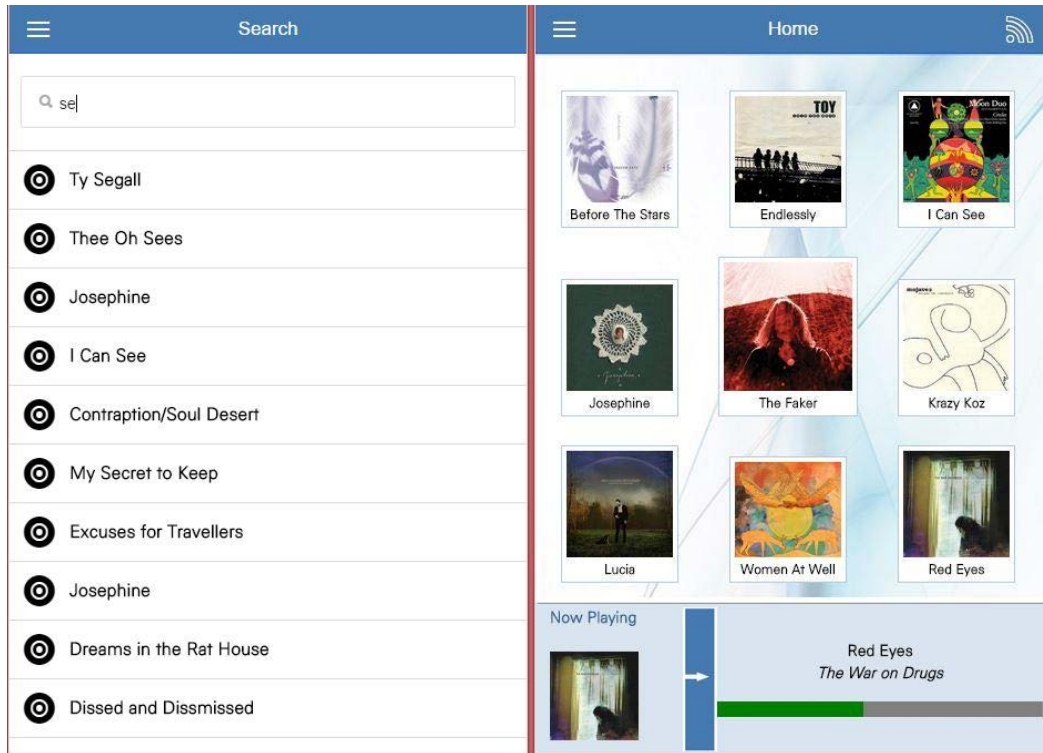
Simple task – Listen to a Soneme DJ. Search through Soneme DJs in the area and choose a DJ's broadcast. This is a fundamental part of the app because it allows listeners to discover music through other Soneme listeners in real time.



DJ directory screen

DJ listening page.

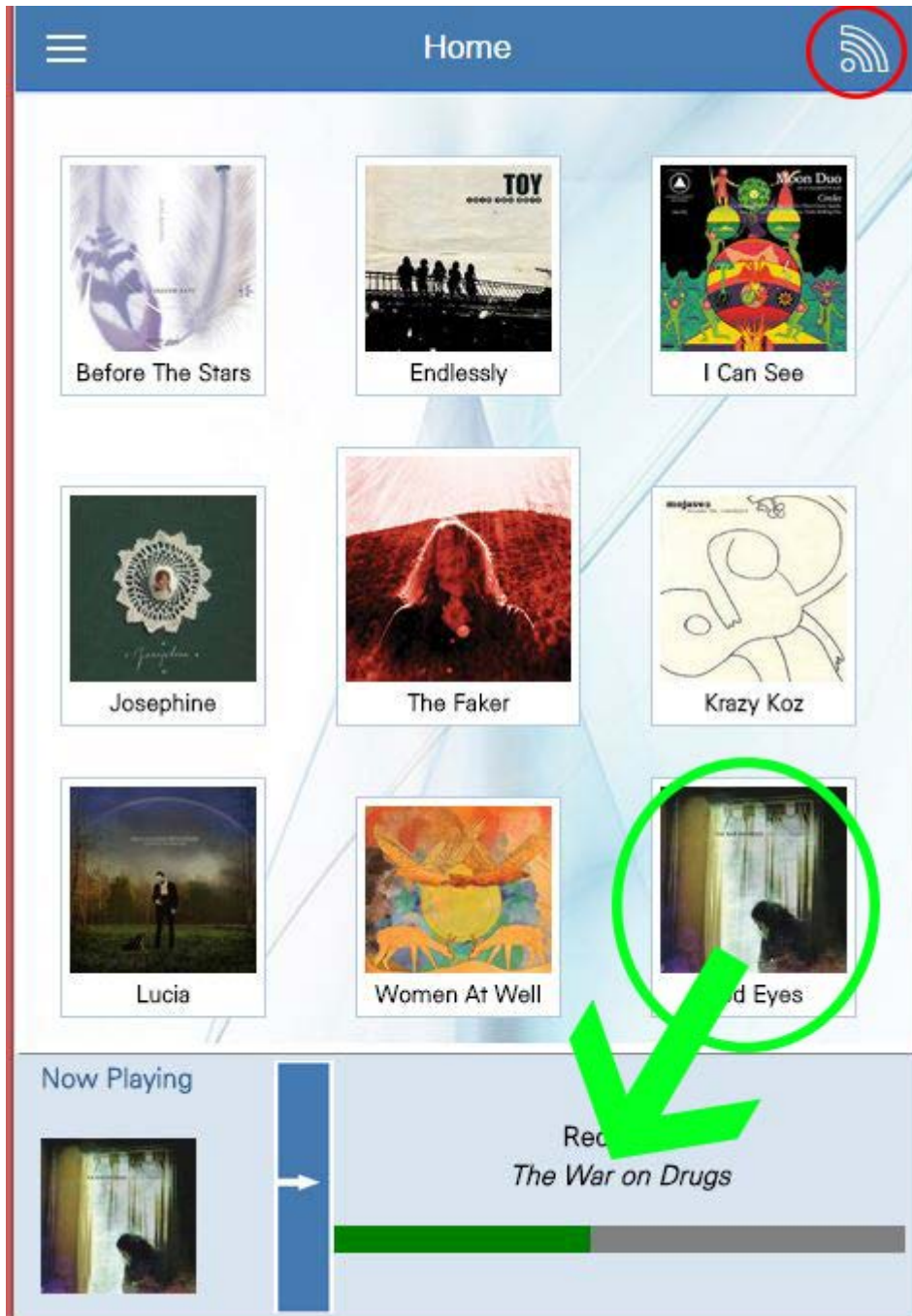
Medium task – Make an initial search for songs to listen to. This is the very first thing a user does when starting Soneme for the first time. From then on, the search function is always available from the main menu, so a listener can frequently create a new music grid at almost any time by making a new search. The search process can be somewhat sophisticated because a user can specify within several search categories (genre, style, artist, etc). From the search, the grid appears and a listener can start exploring music.



Search page

Music grid page

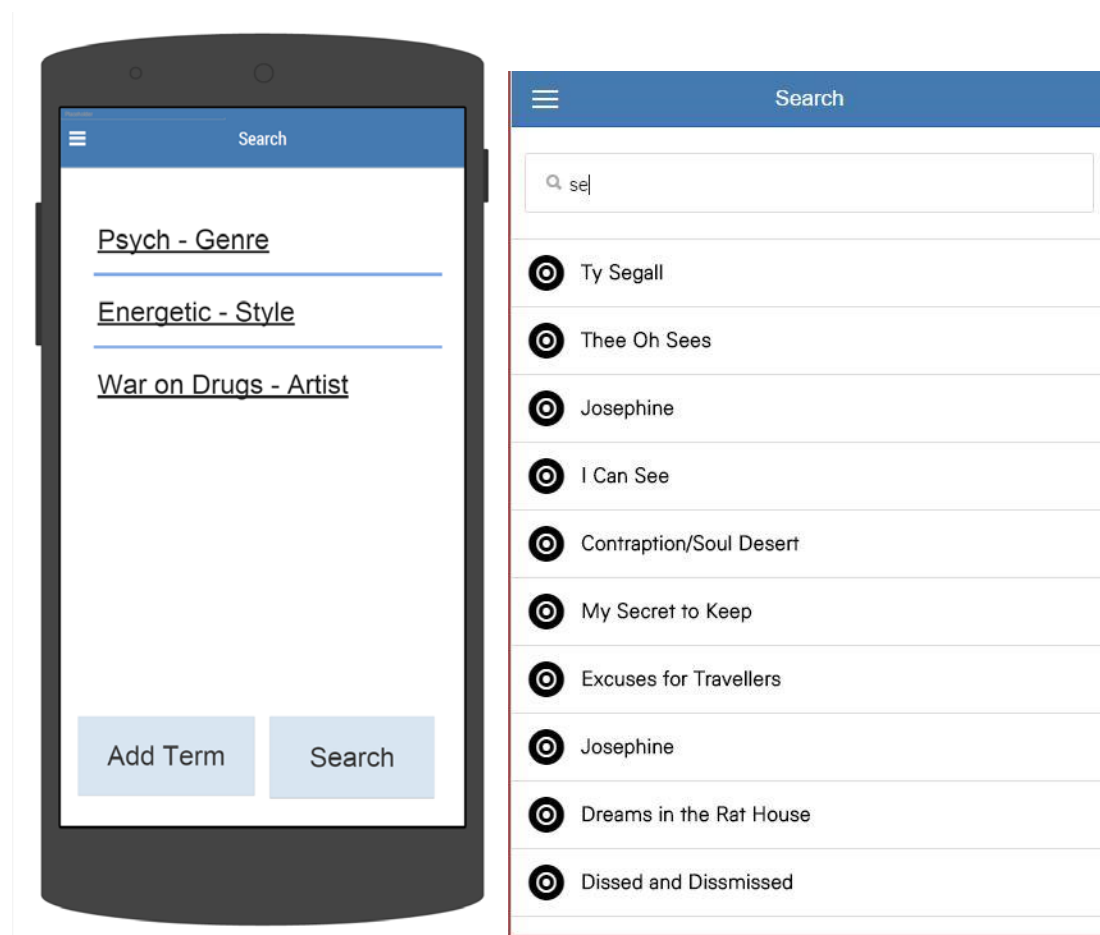
Complex task – Become a Soneme DJ. Add and remove songs from your queue (green arrow), reorder those songs, and broadcast your music selections (tap the icon in the red circle). This involves a user dragging items between the queue and the grid, arranging songs, and searching for new songs in the grid. These actions allow listeners to share music with others in real time, as an analogue to a DJ doing a music show on the radio.



Major Usability Problems Addressed

Most of our medium-fi prototype's heuristic violations stemmed not from our intended design but from the limits of the prototyping process—especially the limits of proto.io, our prototyping application. As we began to reconceptualize our app for the medium-fi prototype, we decided to make drag-and-drop a significant action within our app. However, we could not implement drag-and-drop within proto.io, so our prototype was more realistic from a visual standpoint than an interactive one.

Similarly, our “Search” feature depended on a level of interactivity not available within proto.io. Finally, to streamline the prototype design we implemented the ability to get more artist info for only one song, even though we plan to make that feature available for all songs. Our design changes based on the heuristic evaluation, then, were generally not overhauls of the UI but instead smaller tweaks.

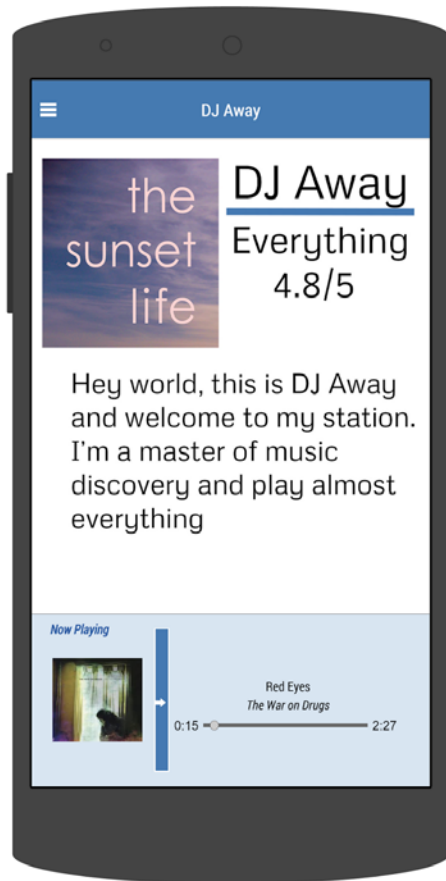


Medium-Fi search screen

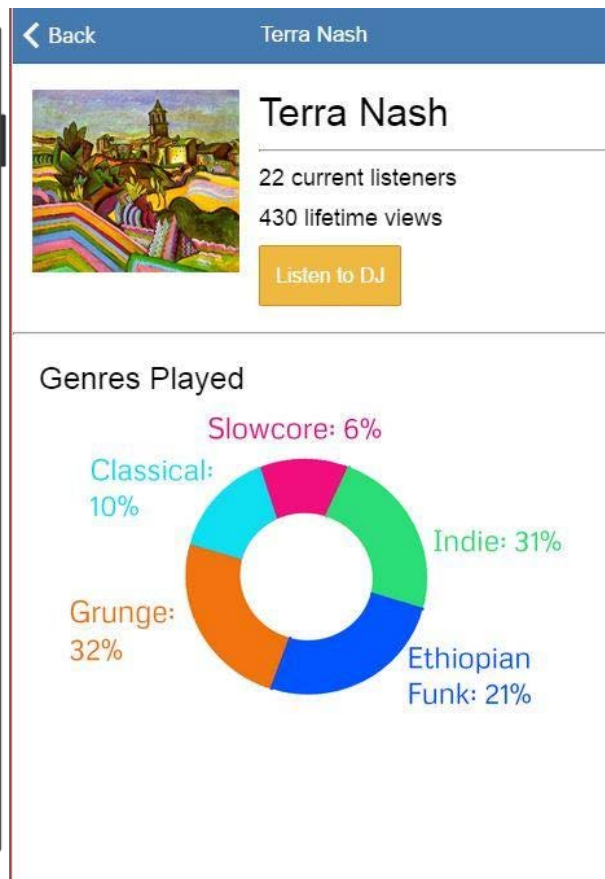
Final search screen

We initially proposed a feature for rating DJs, but our implementation of this feature confused the evaluators. Our rating system in the medium-fi prototype was a number (with decimal points) out of 5, which could have been substituted with a five star system for standards' sake. But this also brought up the problem that certain DJs would eventually have the majority of listeners, creating the kind of popularity contests that this app was designed to avoid. We decided to

eliminate the DJ rating feature altogether and replace it with a more useful metric: genres commonly played by the DJ.



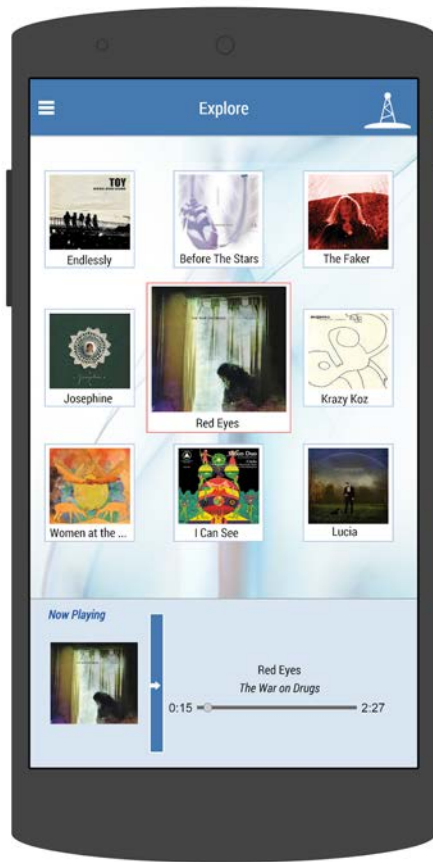
Medium fi DJ page



Final DJ page

Our off-screen menu received criticism that it required an extra tap, and that it had too few options (four) to be space efficient. It is true that the menu is currently light on options, but we think there are still enough to warrant some kind of separate menu—especially with the view that additional options might be enabled in the future. We toyed with a more space-efficient drop-down menu, but because of concerns about menu standards on apps (as well as time constraints), we've kept the off-screen menu in the high-fi prototype.

Another objection is that the grid contains only nine items at a time. The suggestion was to allow listeners to scroll through a much larger grid. This could potentially work if pulled off right, but this brings up the issue of choice overload. While nine items is relatively restrictive, the number of options increases constantly with each new menu; it's just that the options are streamlined in a way that doesn't make the selection process immediately overwhelming. Additionally, in our conception of Soneme, the grid is not static. So we've kept the nine-item grid.



Medium fi grid screen



Hi Fi grid screen. Few changes from the medium fi prototype.

Evaluators suggested our tutorial could be more helpful. In the medium-fi prototype, the tutorial was limited to one screen at the first opening of the app. In part, this was meant to be a stand-in for a better, more pointed tutorial; the medium-fi prototype made it difficult to include hints that would appear only once within the prototype. Although our UI is relatively unconventional for a music app, we've taken steps to keep it simple, so the need for a tutorial would be relatively minimal anyway. For the purposes of simplicity and presentation, this version of the prototype did away with a within-app tutorial.

Evaluators expressed confusion over the purpose of having the center album cover in the grid be larger than the ones surrounding it. Their concern had to do with keeping size standards, so uniformity of objects in the grid. Additionally, they weren't necessarily sure whether this meant the song in the center is also the currently playing song. We've chosen to keep the album cover in the center larger. The size difference is supposed to signify a kind of "if you like the song in the center, you should also check out the songs in the periphery"—an outward radiation of recommendations. Also, a more interactive prototype than the one we could make in proto.io would make clearer that info about a currently playing song is at the bottom of the screen. However, in the hi-fi prototype, we did make the "Now Playing" larger so the distinction between the "Now Playing/Queue" section of the app and the grid is clearer.

Finally, a concern was raised over the action required to bring up the Artist Info page for an album cover. The evaluators wanted it to be more intuitive. In the hi-fi prototype, a listener need only tap on an album cover to access the page.

Moving beyond the heuristic evaluation, we made a few further changes to the UI. We changed the DJ Profile pages to make them more distinct from the Artist Info pages. Specifically, the DJ profile pages contain less text, but they now contain percentage statistics of the music genres a DJ plays, represented as a wheel with different genres corresponding to different color bars within the wheel.

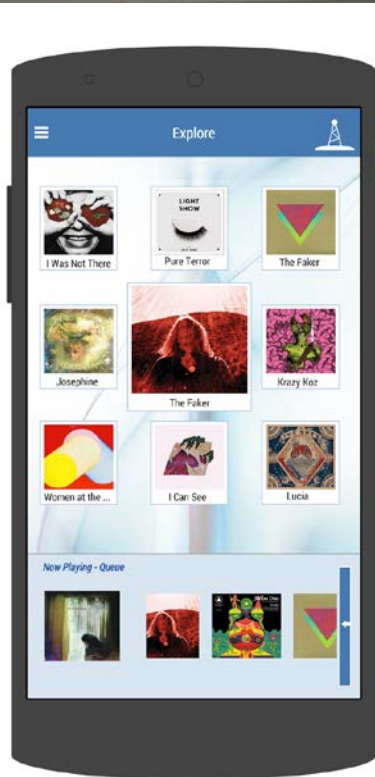
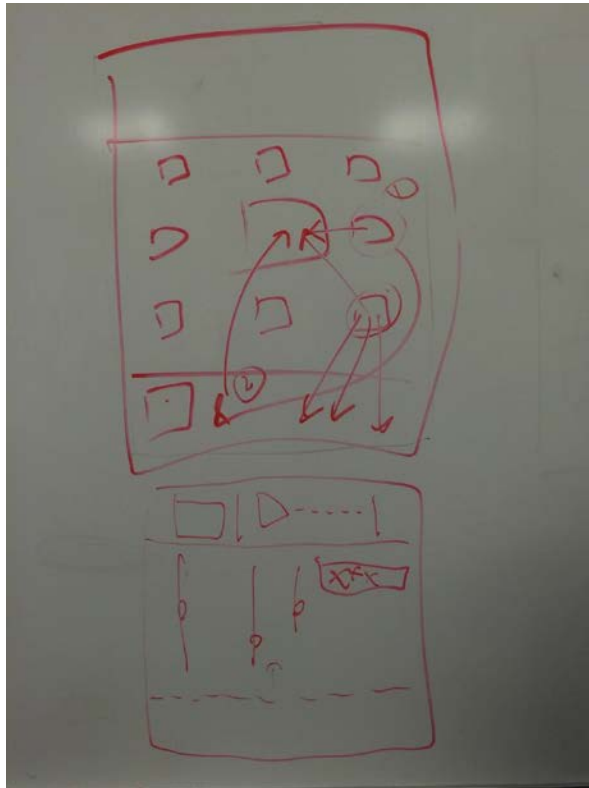
The search screen for this prototype now has only one search field, and we've created different search categories. A listener can make searches based on the name of an artist, album, song, genre, mood, or time period.

Design Evolution

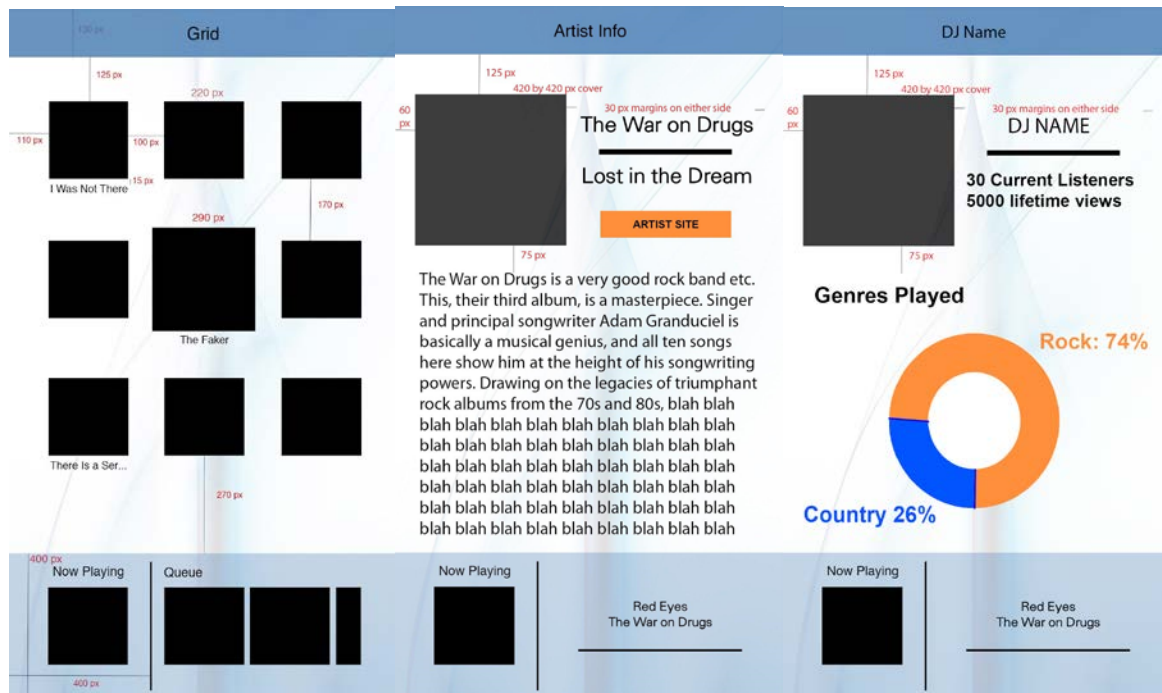
At the very beginning, Soneme consisted of just one idea: a music app that boosts lesser known music by making popularity a metric when giving recommendations to listeners. The idea was vague at this point, but it stemmed from the desire to help listeners break out of recommendation cycles found in apps such as Spotify, and to help connect listeners to great, lesser known artists (often meaning artists who sell only dozens of albums).

We had many ideas for a UI early on. These ranged widely and included “webs” of music (networks of recommendations that the app could visualize for listeners), or “maps” (different paths that would be presented like a video game and offer different music recommendation). The idea of a music grid, though, struck us as especially interesting and useful. By having recommendations radiate outward from a certain song, the grid offers a means of showing connections between artists/songs/genres/etc, but it presents them in a way that isn't overwhelming. A few options appear at one time, but they give rise to new options, and so the number of potential paths grows exponentially.

Soneme's evolution has followed a clear pattern of simplification and refinement. The initial vision of Soneme consisted of numerous functions that sometimes did not form a coherent whole. This became apparent when we had volunteers perform usability tests on our low-fi prototype. For example, at the beginning we initially had “sharing” and “broadcasting” functions that were completely separate. In another place, the “queue” function was represented in a confusing way. Additionally, the profusion of buttons in our low-fi prototype added to the confusion. Simplification began in earnest with the idea to use DJing as a more consistent metaphor for the app, and with the accompanying idea to use a drag-and-drop paradigm in the grid screen. Dragging and dropping allowed us to eliminate confusing icons on the grid screen and also gave the interface a dynamic and interactive feel.



Top Left: Concept sketch. Top Right: Low-Fi prototype. Bottom left: Medium Fi prototype. Bottom Right: Hi-Fi prototype.



Mockup screens for the final prototype showing exact pixel dimensions.

For our final prototype, we kept our overall paradigm, but focused on making the interface as readable and usable as possible. We took the time to figure out exact pixel dimensions for each UI element and made sure our fonts were large and readable. The results from our heuristic evaluation suggested that some elements were not clear enough, and in our own usability tests we came to the same conclusion. Fonts from our medium fi prototype were too small for mobile use, album names were hard to read, and icons such as the side menu and broadcast button were tiny. We took all of these elements and increased their size. The heuristic evaluators also noticed that some of our gestures were overly complex. After review, we decided that we had become overzealous with mapping actions to drag-and-drop interaction. We reduce drag-and-drop to only the areas where it made sense. For example, our medium fi prototype allowed users to read artist info by dragging the album cover to the top of the screen. Testers found this interaction confusing, so we changed it to a simple tap.

Prototype Implementation

We began with the intention to develop our app specifically for Android. We created the initial project and started developing, but very soon we realized that it would be much harder to get the prototype done in two weeks than we anticipated. So we opted to go and develop a mobile web prototype, which could be packaged up and deployed on a mobile interface.

The prototype was developed using the angular.js and ionic frameworks. Angular provides a structure which allows easily faking data to look like it's coming from an API, and handling the

views and navigation. Ionic provides UI frameworks which allows simulating mobile interfaces on the web. With these libraries together, it was easy to make the initial setup.

The biggest challenge in developing the prototype was integrating the drag and drop functionality. For it we used a user developed angularjs drag and drop library. After many trials, we got it integrated, however it was hard to make it work appropriately with ionic UI. There were many hacks necessary to get it working. At the end there was one unresolved problem around a conflict between the vertical scrolling behavior and drag and drop functionality. This made it such that touch didn't allow drag and drop to work, only with mouse. This type of functionality would probably have been easier to perfect on mobile, but at the end, it was easier to get the basics on web.

The search terms on the search list was a list of 100 different songs/artists/genres hardcoded into the system. Yet the way it functioned and searched was actually real.

When searching and selecting something, it always showed the same results, so this was more of a navigation simulation than anything. We hardcoded 25 different albums into the prototype. Therefore the songs which came up whenever a new song was placed in the center were just random. However, the placement, drag and dropping, and songs being played on the queue were accurate to what was said to be playing.

We didn't spend too much time on making the playback of too many songs work, since it seemed like most people who look at the prototype don't spend enough time to listen to many songs. Since the songs can't be changed (due to the fact that you're broadcasting), it didn't seem worthwhile to give priority to multiple song playback.

At the end of the day, the whole UI, feel for the application, and interactions was complete. The only pieces missing where around the playback of the correct music, and plugging in the correct algorithm to make songs actually match.

However, people trying the app don't really know if the songs being shown are actually related (since it's lesser known music), so it seems safe to say that it was as accurate as possible.

If we had more time, we would definitely concentrate on making the playback experience work, especially the broadcasting and listening to DJs component of it (currently simulated).