

soneme

Medium-fi Prototype Report

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

For the moment, we've eliminated a share function from our prototype and our tasks. During our low-fi tests, we found that we didn't have a clear conception of how sharing music from Soneme to social media should work. Otherwise, the tasks remain similar.

(The tasks below are ordered in a way they'd likely be performed.)

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. The search process can be somewhat sophisticated because a user can specify several search fields as well as several search categories (genre, style, artist, etc).

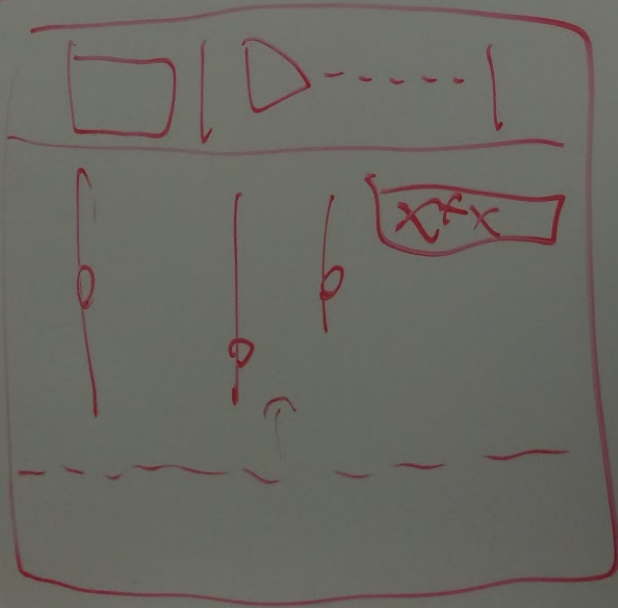
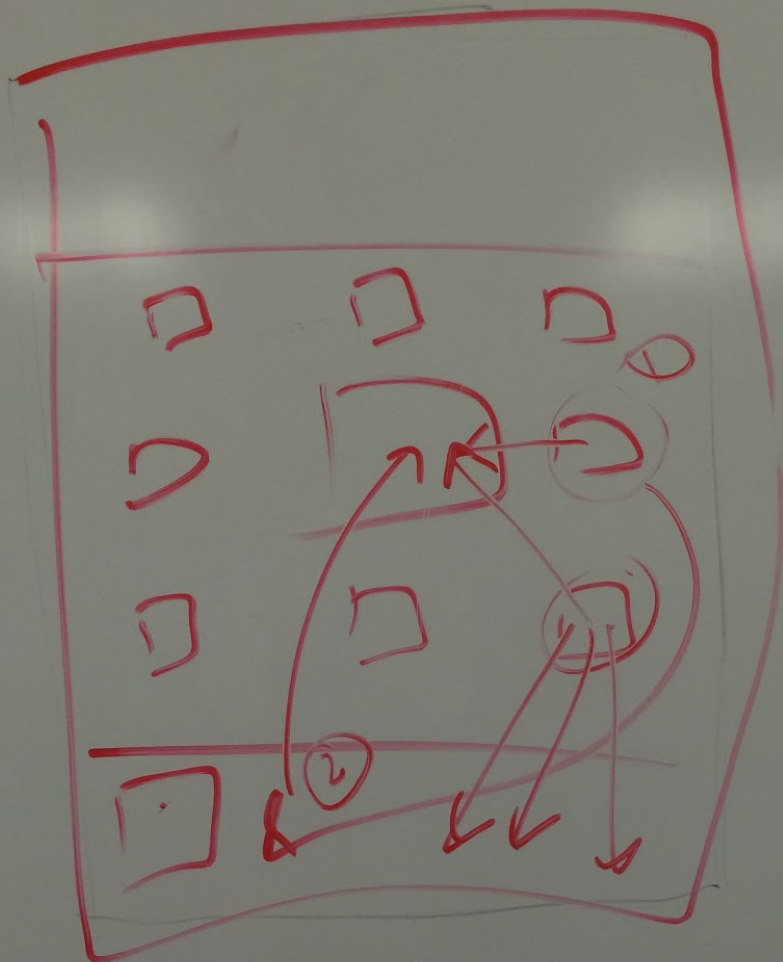
Complex task – Add and remove songs from your queue, reorder those songs, and broadcast your music selections. This is the curator/DJ side of Soneme and involves a user dragging items between the queue and the grid, arranging songs, and searching for new songs in the grid.

Simple task – Search through Soneme DJs in the area and tune into a DJ's broadcast.

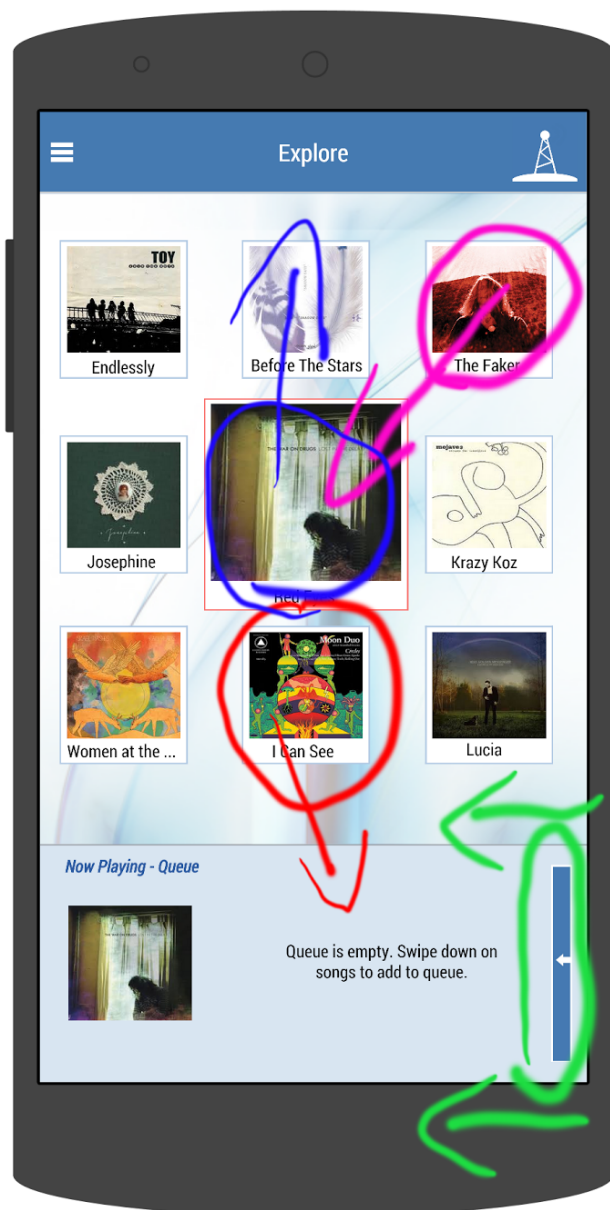
Revised interface design

In our low-fi prototype, we were still in the early stages of designing an interface around app-based DJing. Our original idea of "playlists" gave way to a simpler "queue" where a user could store songs to play later. But this initial idea of the queue—revealed by a small button near the bottom of the screen—gave our low-fi testers difficulty. How could we make the function of queueing more visible? Instead of a button, we decided to make the queue hidable behind the "now playing" tab, which could be easily retracted

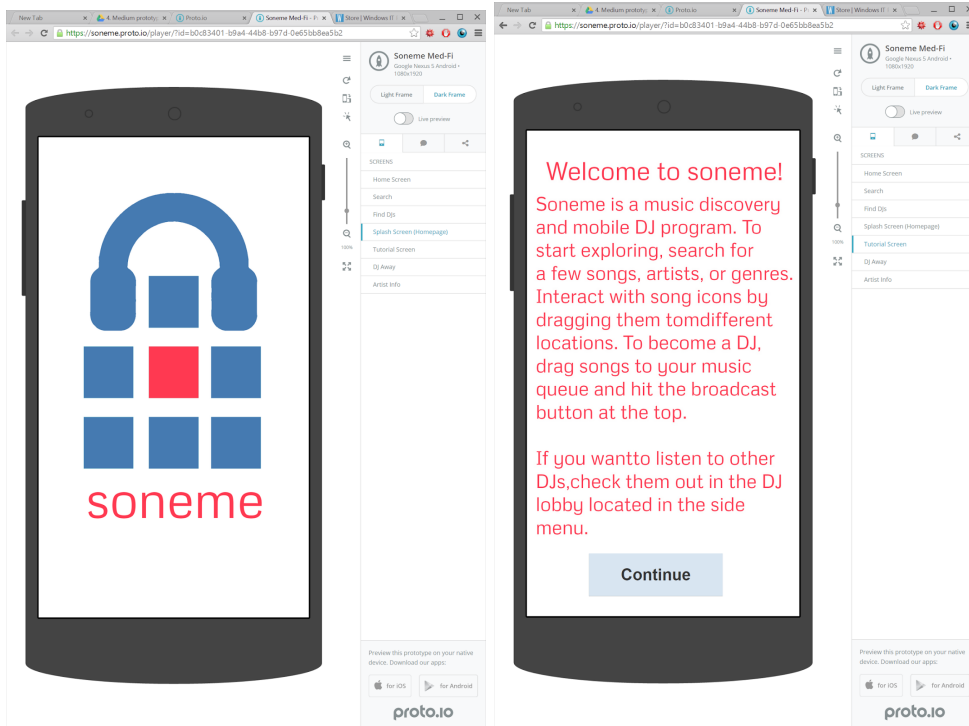




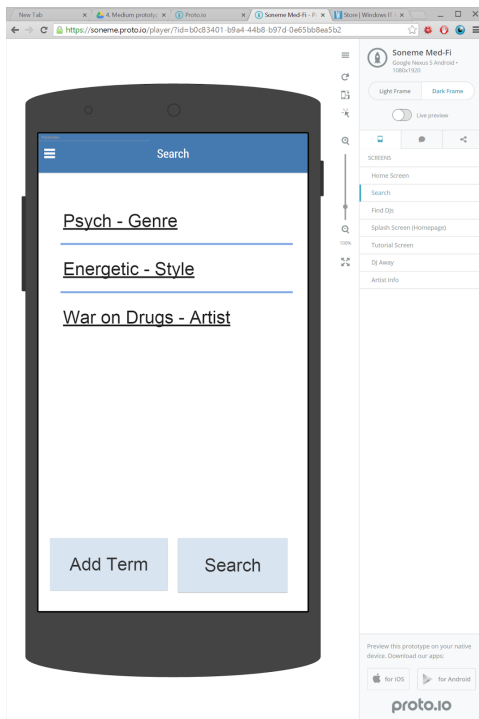
Once we removed the queue button, we realized that we could streamline the interface even further and let users access features through touch gestures. The testers from our lo-fi prototype naturally tried swiping motions for the grid screen, as it is already a common touch screen convention. We redesigned our grid screen to use many swiping gestures in place of buttons. In our new interface, we replaced the info button with an upward gesture on the center square. Instead of holding down on a song to bring it to the center, users now drag it. The queue button has been replaced with a slider, and users add, remove, and move songs in the queue by dragging them into the queue, up out of the queue, or left and right to change order.



Medium scenario—Soneme starts with a splash screen and then a short tutorial.

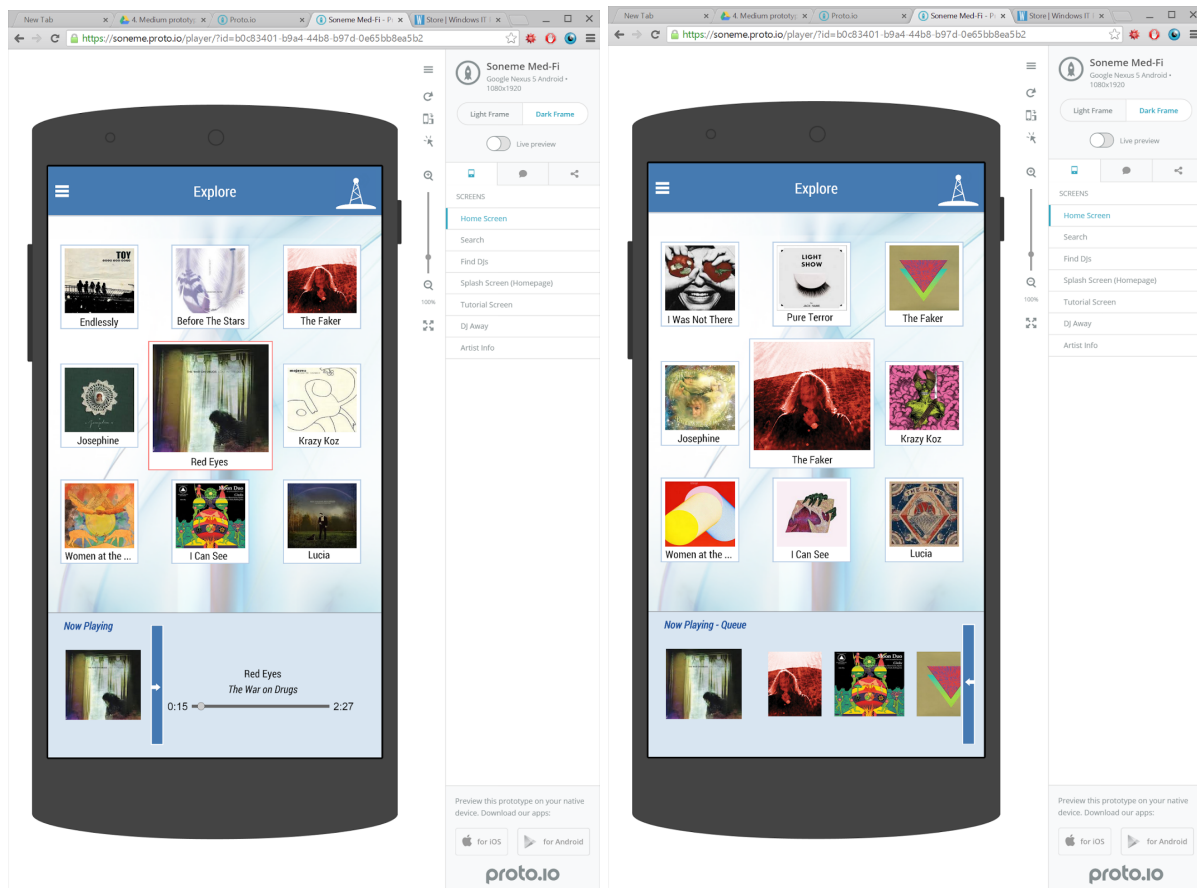


The “Search” screen appears, and the user can type several search terms of different types. This screen is also accessible from the menu.

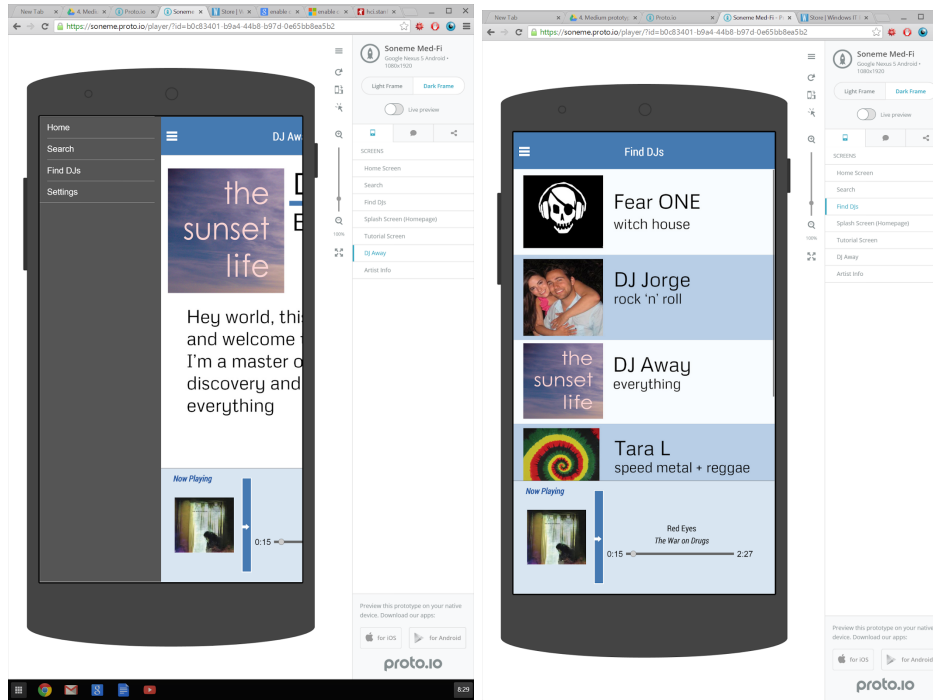


Complex scenario—Your music is now playing. Tap on the vertical blue stripe at the bottom of the screen, and the “now playing” info will disappear and reveal your queue. (In a later version you’ll swipe the “now playing” box to the right.)

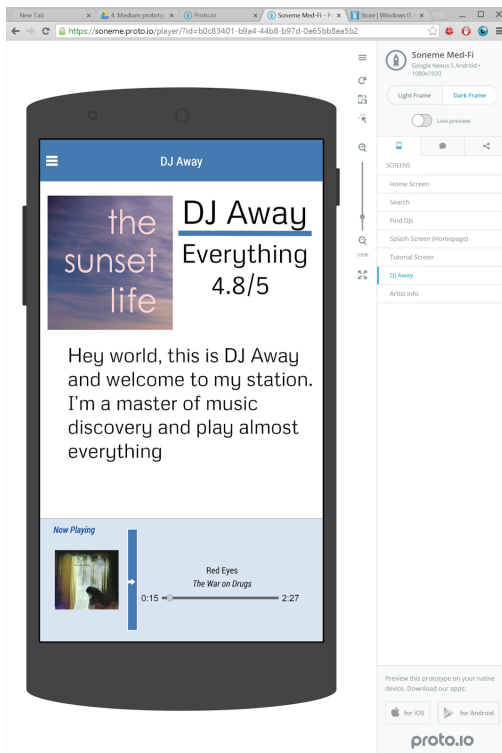
Add a song to the queue by dragging that song’s album cover from the grid and down to the queue. To remove a song from the queue, hold the corresponding album cover on the queue and swipe down. (This removal part hasn’t been implemented yet.)



Simple scenario—access the “Find DJs” option in the menu and scroll through the resulting DJ list.



Select one of the DJs to start listening to their broadcast.



Prototype overview

We used proto.io to build this prototype. Proto.io is a feature rich online prototyping tool for mobile platform. We found that it was easy to put together screen layouts, since the tool contains numerous pre made elements that we could include. It also has a very complex interaction system to link the pages to each other. While the interactions were numerous, we found that it was missing some important features to achieve our vision. Our UI vision has become heavily dependent on drag-and-drop actions, so we wanted to implement a prototype that matched our concept. Unfortunately, proto.io is not sophisticated enough to enable drag and drop interactions. We substituted directional swiping for dragging, which was an acceptable substitute.

We also discovered that proto.io is great for creating very simple prototypes, but tedious to use with a prototype with a lot of interaction. To create a fully functional music grid, we would have had to statically design every single grid screen and add a link for every piece of album art. Due to time constraints and lack of dynamic control, we implemented interaction for only a few songs.

When simulating the search screen, we decided to statically define the search parameters and not let the user type in a query. We assume that mobile users have tapped a search box and entered a query at some point, so testing the usability of a non-novel interface component was not important. Additionally, any searches would have been limited anyways, since there is no back end to provide search results.