

Vesta

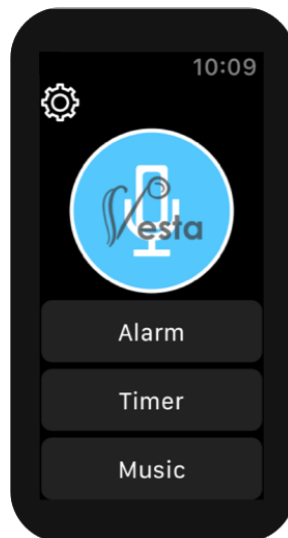
Start every day on a high note

Allen Zhao ≡ Development and User Testing
Mohana Moorthy ≡ Product Management and User Testing
Yinglan Ma ≡ Design and Development

Problem and Solution Overview

These days, many people spend hours at a workplace, only to come home drained and exhausted. Our goal is to bring back energy into everyone's home lives. Vesta is a voice-controlled, intelligent audio assistant that motivates you using audio - it wakes you up, encourages you to do your household chores, prepares you for events (such as meetings or dates), and even plays music based on your mood.

Why not add background music to your life? Be the star in your own movie.



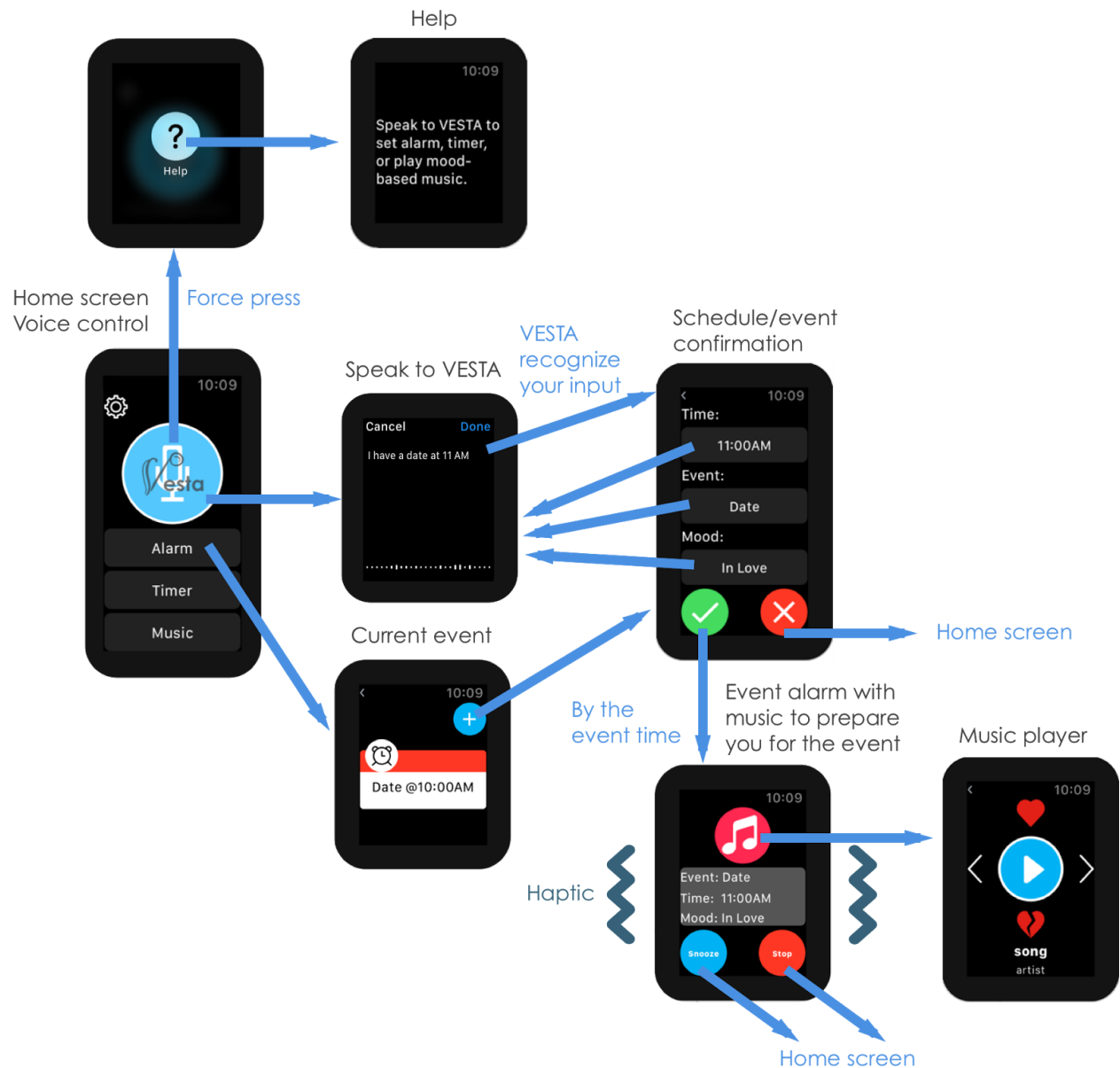
Tasks and Final Interface Scenario

Task 1: Simple Task - Automatically suggests and plays music based on mood.

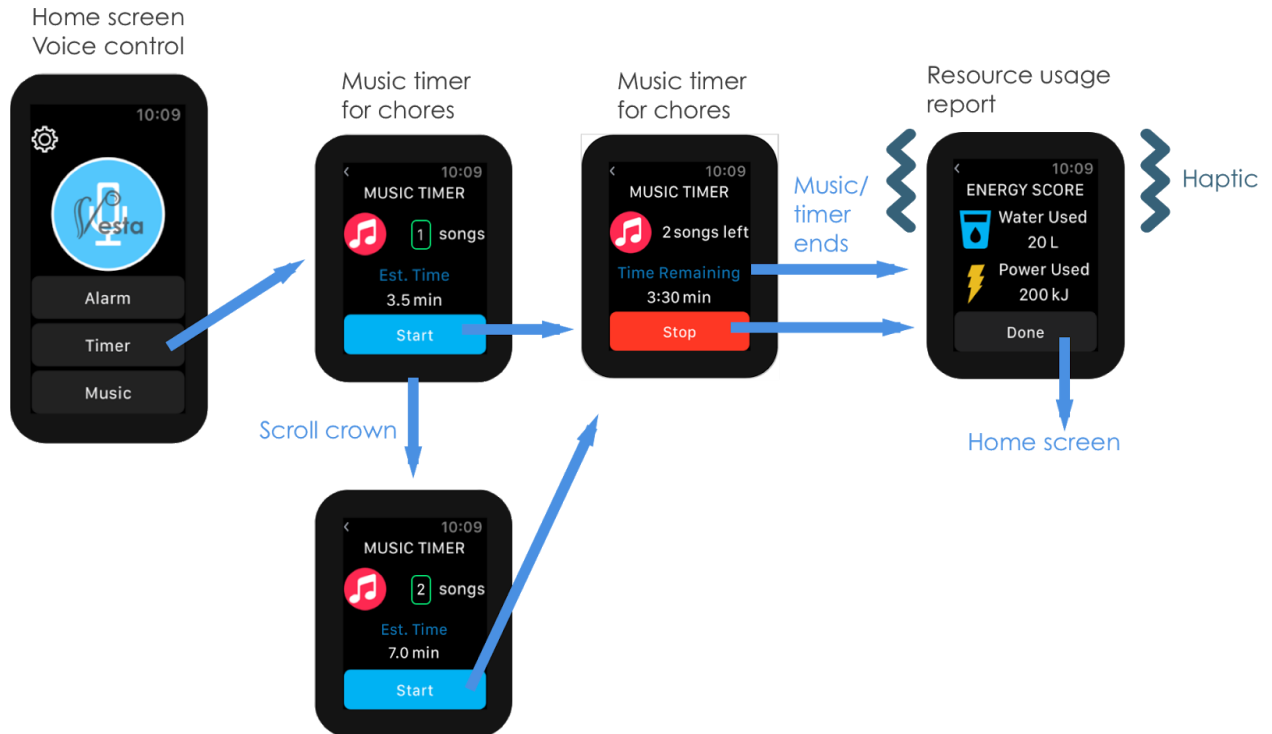
One of the biggest advantages of building a watch app is the access to heart rate and other features that indicate mood. We chose this task because it's a great way to motivate the user through audio. Users can either directly ask Vesta to play music that fits their current moods or play it through the music button.



Task 2: Moderate Task - Set an alarm for an event with vocal input. Then, listen to the fitted music to get ready for that event. We wanted our assistant to be relevant in all parts of daily life and expected an alarm function to be heavily used (based on user feedback), so decided on this as our moderate task.

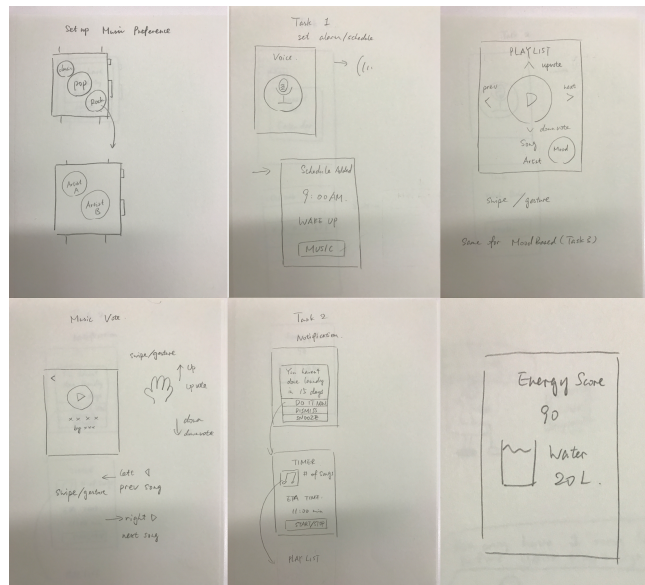
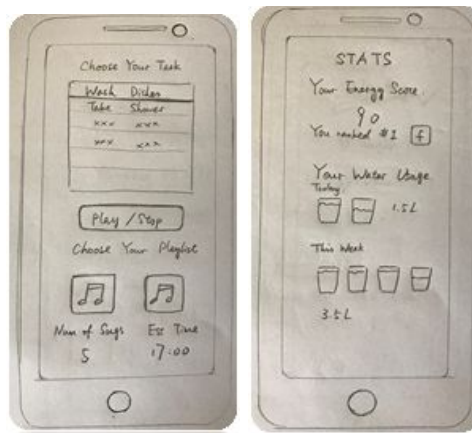


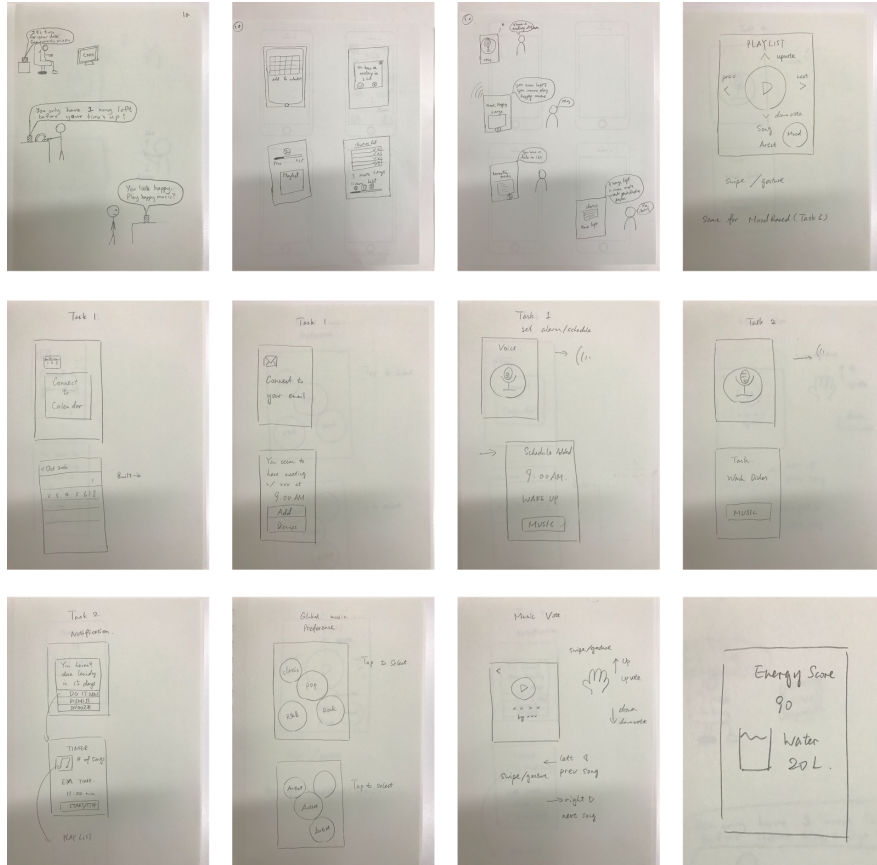
Task 3: Complex Task - Track the time spent doing household chores (like dish washing) by listening to music (each song is roughly 3.5 min long) while monitoring the energy footprint. This feature was chosen for 2 benefits that it offers users: a more fun chore experience and a way to promote conservation.



Design Evolution

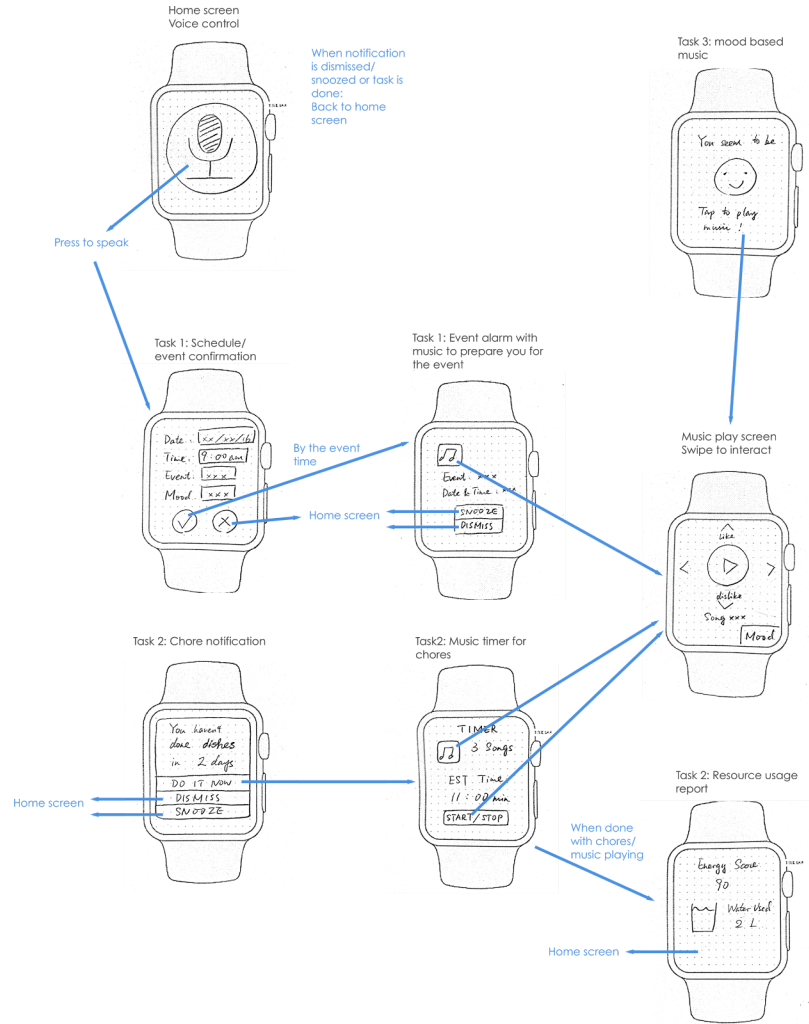
Our initial sketches, while very crude, allowed us to get a good idea of how users would use our app. In particular, they showed us that users didn't want a lot of buttons to push while they worked on chores; we also found that users didn't want to bring their phones out very often during menial tasks. As such, we decided on a smartwatch app for our Vesta platform.



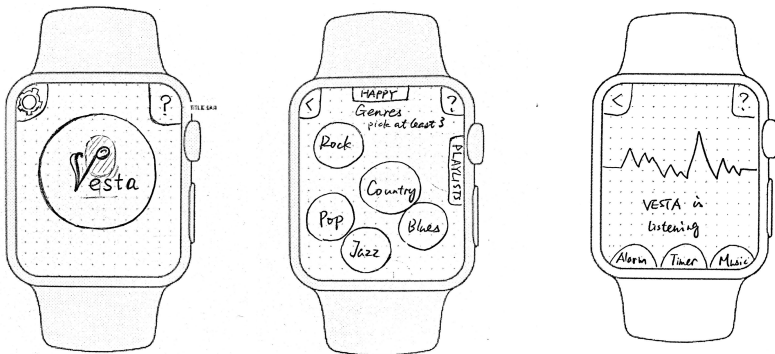


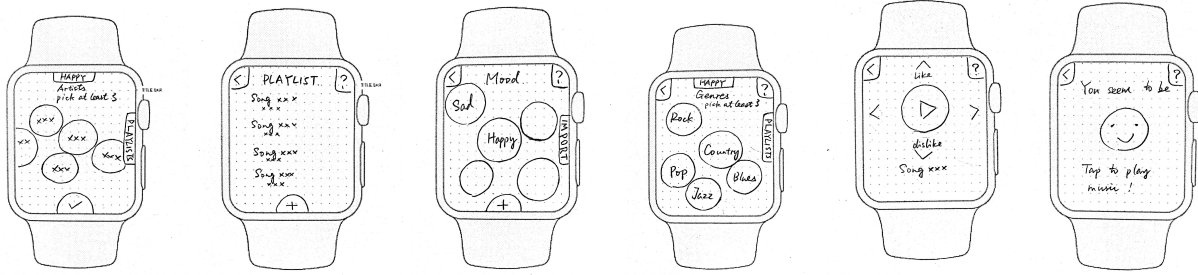
Low-Fi Prototype

With our initial low-fi prototyping, we iterated on a smartwatch prototype design until we had ironed out critical issues with users - which led to us adding a 'like/dislike' button to the music screen and a indicator for when Vesta is 'listening'. We noticed here that smartwatches don't have keyboards, so we added voice recognition to carry out all possible tasks in the next screen. Additionally, we installed back buttons on every page of the app and simplified the music screen.



Low-fi Prototype Revisions:

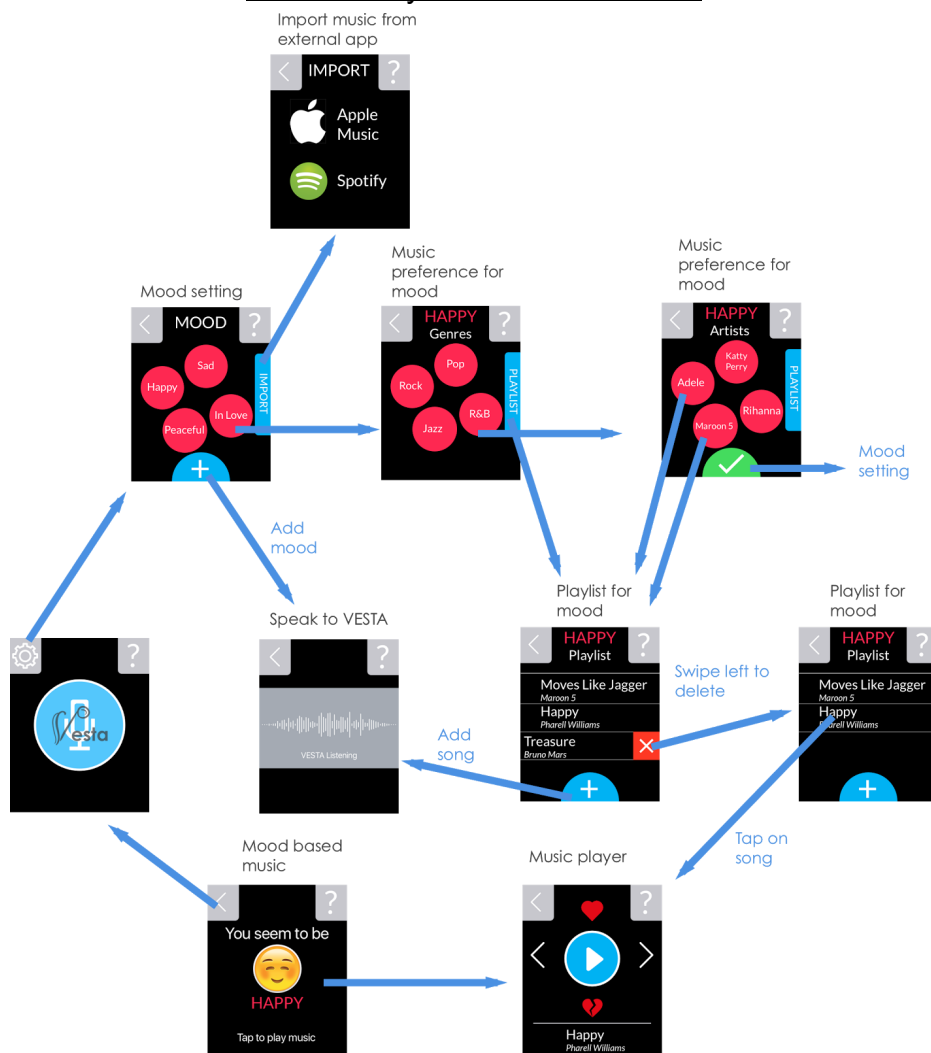




Medium-Fi Prototype

For our medium-fi prototype, we implemented the UI design from our low-fi prototype in Marvel. The only change from the revised low-fi stage consisted of changing the 'like/dislike' buttons to 'heart/broken heart' since images are more intuitive than text. Following this, we used heuristic evaluations to get to our final prototype.

Task 1: Play mood-based music

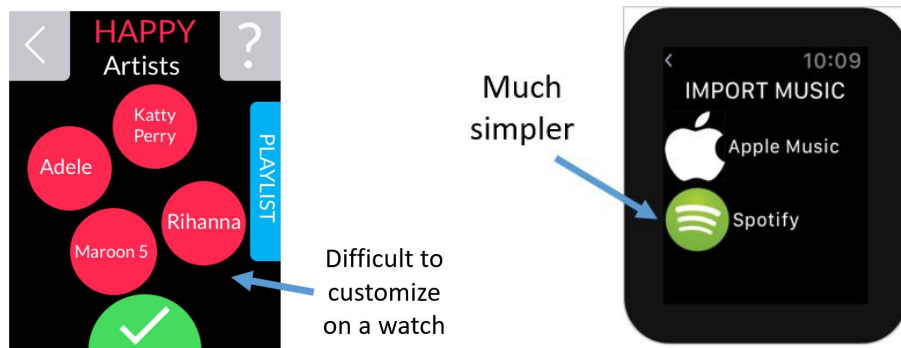


Major Usability Problems Addressed

Note: Images on the left are medium-fi prototypes with the issues, while images on the right are the fixed hi-fi prototypes

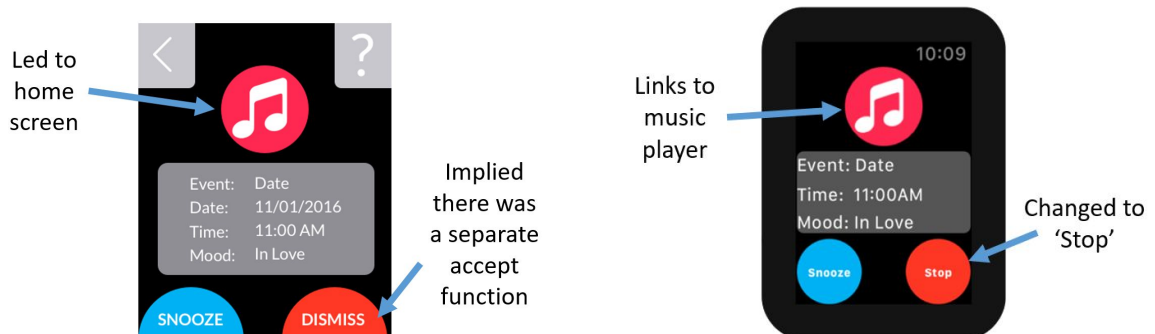
- **Problem:** Task flow surrounding the mood-based music is very confusing and clunky. The pages where the user sets his preferences for different kinds of music by picking the mood and then the artists seemed unintuitive and too complex for a watch app. (level 3 and level 4)

Solution: We cut out all the options where people can customize their playlists and made the interface simple. Users can import their music from Spotify or Apple music, allowing us to remove the confusing screens entirely.



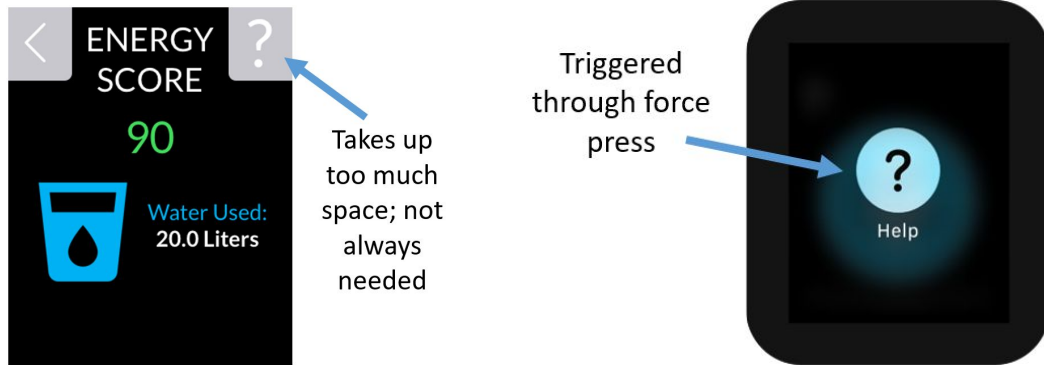
- **Problem:** Event notification had clickable music icon to return to home page; this was confusing. The “snooze” and “dismiss” buttons led to issues with people that tried to accept the event by clicking on the music button, which just took them to the homepage. (level 3)

Solution: The music icon on this page now leads to the music player screen, which makes far more sense than the home screen. We changed the buttons to display “snooze” and “stop”.



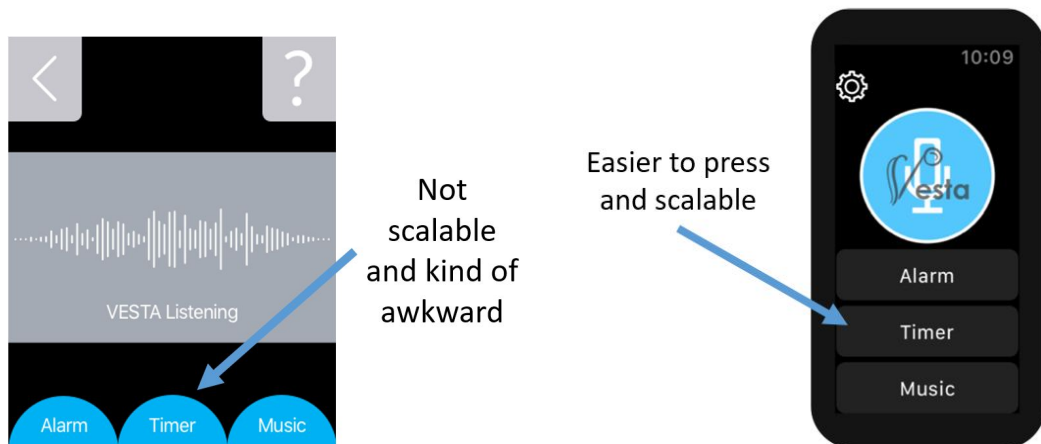
- Problem:** The help button was present in every screen and took up a lot of space. It doesn't make sense to waste valuable screen space for something which people won't use once they are comfortable with the app. (level 3)

Solution: We added the ability to force press to invoke the help screen.

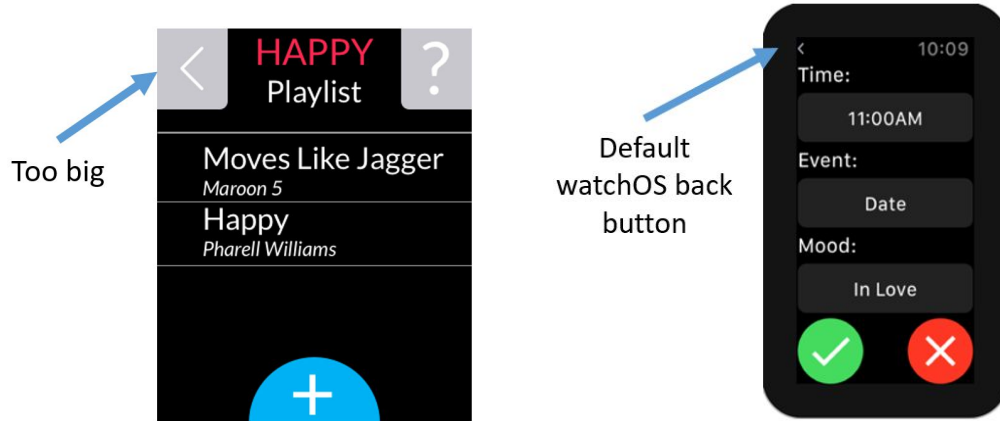


- Problem:** Task icons were clunky semicircles on the bottom of the 'listening' screen. They were somewhat difficult to press and this format doesn't scale well with more tasks due to restricted space on the bottom edge. (no level; this was a team-driven change)

Solution: We moved the task icons to the home page and stacked them below the voice input button; this makes it easier for users to access and allows for infinite tasks..



- Problem:** The back button we designed took up too much screen space and were missing on some crucial pages. (level 1)
Solution: We used the default WatchOS back button instead of our design and added it to every page.



Prototype Implementation

We implemented our high-fi prototype using Xcode 8 with Swift 3 and WatchOS 3 WatchKit extensions. Our build target was the Apple Watch Series 2 38mm. These tools allowed us to build a real Apple watch app with complex user interactions such as voice input, force press, and crown control. We could now provide users visual, sound and haptic feedback. Additionally, we are capable of connecting to a bluetooth speaker to play music for users, which matches the designed context of our app.

However, our prototype is still restricted by the limitations of WatchOS. Notifications do not support long soundtracks, and have limited user interactions, so we used a timer to push the notification screen instead of having a real notification. Gesture recognitions are not fully supported in WatchOS, so we didn't implement them for our music player interactions.

We hard-coded the music imported from external apps and the timer used to fire notifications. Currently there are only two mood categories - "happy" and "in love" - and the chore timer can only play up to 2 songs (since we only had 2). We also used Wizard of Oz techniques to simulate the underlying AI algorithm that classifies suitable music for a given mood, as well as mood detection based on user body temperature/pulse.

The speech recognition works well, but only recognizes 2 events: "wake up" and "date". It maps them to "happy" and "in love" moods, respectively, so a more robust and comprehensive speech recognition system is needed. Finally, the alarm screen only shows one alarm, but we would like to support a list of alarms for the day. Other future

plans would include implementing the underlying AI algorithm that would choose music based on mood and a mood detection algorithm that detects the user's mood from watch sensors.

Summary

Vesta was born as a set of vague sketches that defined our goal of brightening the home lives of people everywhere. Through weeks of design, coding, and evaluations, Vesta has undergone many iterations - constantly improving on feedback from each of our prototype stages. Our final prototype, with its refined interface and functioning UI, now barely resembles the first paper sketches, but our mission statement is still the same. We hope that by building Vesta, we have helped to inspire others (and ourselves) to make the home a more musical place.