# NOVEMBER 2019

## APPLICATION DESIGN USER TESTING

We completed our next user test during this time: the design test, which looked at iconography in the instances that an icon would not be accompanied by text, as well as color scheme options for both light mode and dark mode. While we did not intend to use both light and dark modes unless time allowed for it, we wanted to have both tested and ready as a precautionary measure. This test was done via Google Forms survey. Test protocol and results can be found in further documentation on our website.

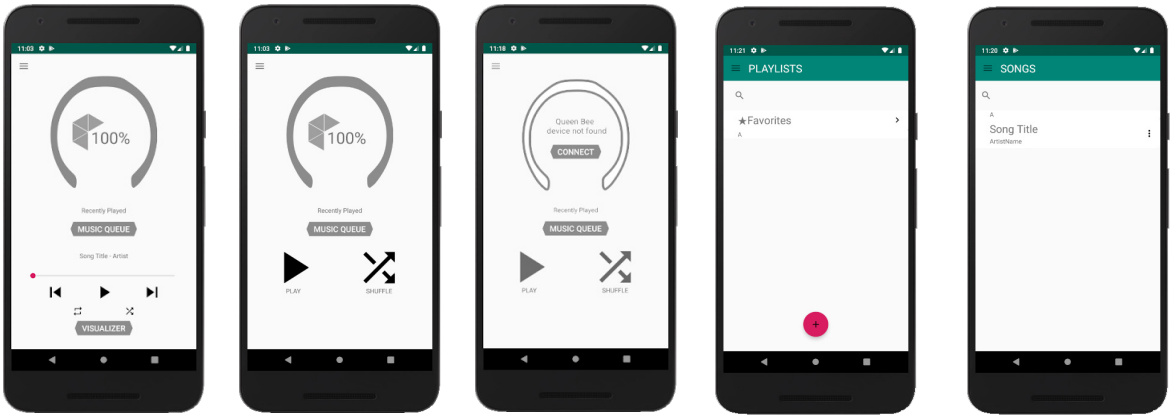
**3D model part additions - Back and vibration motor covers.**

### Prototype 8

- Printer: Ultimaker
- Materials: Prototype 6 with solid black PLA for 'Clip'
- Purpose: Design a method to hold the cover in place on the neckband while keeping the back able to be opened up for alteration of circuitry.

## XML LAYOUTS OF ALL APPLICATION PAGES

Using the high fidelity mock-ups as a reference, along with input from user testing results, each page required in the application was created within Android Studio as an XML layout. Interactive objects such as buttons were placed within the layouts but were non-functioning as there was no logic implemented alongside them.



# NOVEMBER 2019

## BASE MUSIC PLAYER PROGRAMMED

With guidance of online resources, we were able to implement the MediaPlayer library/object native to AndroidX into an Android application. The media player was able to play a preloaded song from start to finish with a button press.

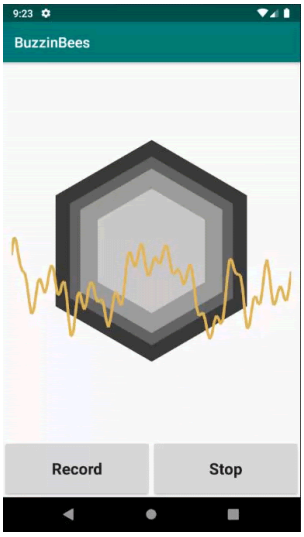
## CONNECTION - APPLICATION & DEVICE'S MUSIC STORAGE MADE

Following Android's documentation and application restrictions, this build required us to implement user permission for data access. With the given permission to read files held on a mobile device's internal storage, the app was able to read and save the audio based media files on the phone.

## CUSTOM LIST VIEWS CREATED - DATA POPULATION OF APP

To properly display user audio files on the music library pages, list view was implemented into the application. A listview is another native element of AndroidX which is designed to host a list of items in a streamlined fashion. Creation of a custom list view allowed us to create a unique layout while maintaining the native functionality provided by Android.

## BASE VISUALIZER PROGRAMMED




The visualizer is a compilation of three Android studio elements: the MediaPlayer class, the Canvas class, and the ImageView class. The MediaPlayer reads and outputs auditory information from the device. This information is then directed to the canvas and ImageView elements which, through use of an algorithm, is able to detect the values of specific frequencies within the audio and translate it into visual feedback.

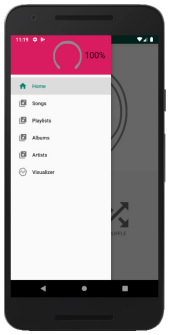
# NOVEMBER 2019

## VIBRATION PATTERN DISCERNIBILITY USER TESTING

Once we were certain our circuitry was finalized, we wanted to test if users would be able to feel the vibrations at all with the motors we chose. We also wanted to see if this varied between different vibration patterns and if users could identify when a different pattern was played. This informed our choice for vibration patterns to use during development. The test protocol and results can be found in further documentation on our website.

**User test report can be provided upon request**

## APPLICATION MENU FUNCTIONALITY PROGRAMMED



A navigational drawer was implemented into the application. This allowed the user to select and navigate to specific pages within the app.

## APPLICATION FEATURE AMALGAMATION

We began on the amalgamation of the various application features we had completed. At this point, this primarily involved implementation of child fragments within the already-present parent fragments. This was done to allow the music to continue playing even on the switching of pages within the application and the allowance of the same MediaPlayer instance in all fragments that needed it for cleaner, more efficient code. Additionally, we began to contemplate multithreading and coding test applications that involved multithreading to get ourselves accustomed to the process for when we would later need it.

## BLUETOOTH EXPERIMENTATION

Following online resources, we continued testing our connection with experiments in sending data across the connection we had established. This proved problematic, in part due to issues discovered in December 2019 regarding chip issues. These issues will be discussed later, but this led to many Arduino code re-writes and the burn-out of a few chips, necessitating replacement chips..