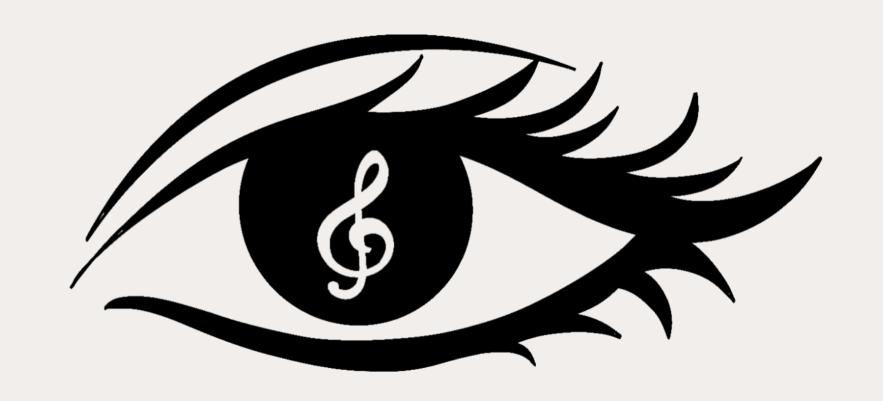


MusiWrite

 \longrightarrow

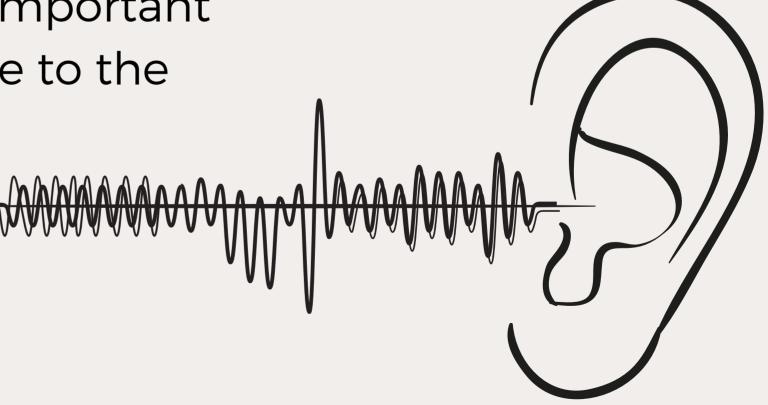
RONIT AVADHUTA
DAIWIK PAL
ALINA SHKURIKHINA



Motivation

 People who are hard of hearing are unable to fully experience music because of their lower audio sensory ability

 Many of these individuals are left out of important parts of their culture and modern life due to the disability



PEOPLE PLACED THE ABILITY TO

HEAR PEOPLE INDIVIDUALLY,

2

IN A GROUP, AND LISTEN TO

3

MUSIC WITHIN THEIR TOP

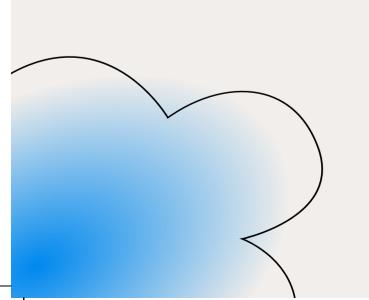
THREE WANTS

76%

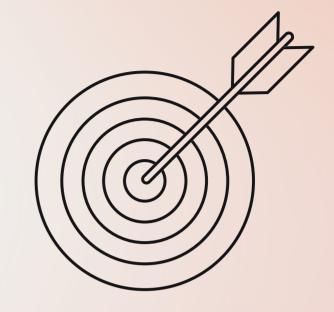
OF PARTICIPANTS BELIEVED

THAT VISUALS ENHANCED

USER EXPERIENCE



Target Audience



PARTIALLY DEAF AND DEAF INDIVIDUALS
RANGING FROM CHILDREN TO ELDERS
WHO ARE INTERESTED IN MUSIC

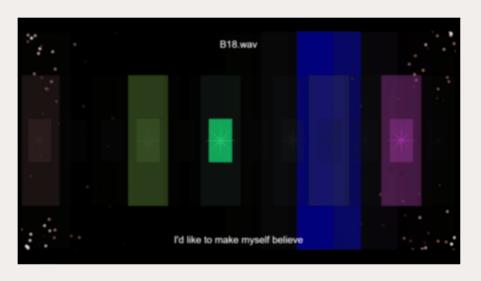
Competitors



ViTune



aims to enhance the music experience using a piano scroll effect that complements the music



Flaws:

- no ability to search music
- can not save music and create playlists
- evokes little emotion from users





iTunes Visualization

creates an aesthetically pleasing visualization that resembles electrical particles correlated to the audio

Flaws:

- little information on how visualization functions
- does not perform well in translating the song's emotion visually



Song Selection

The App provides the user with a list of songs to choose from (prepared MIDI files) and will eventually be able to take in audio from a MP3 file

Description of Solution



Visualization

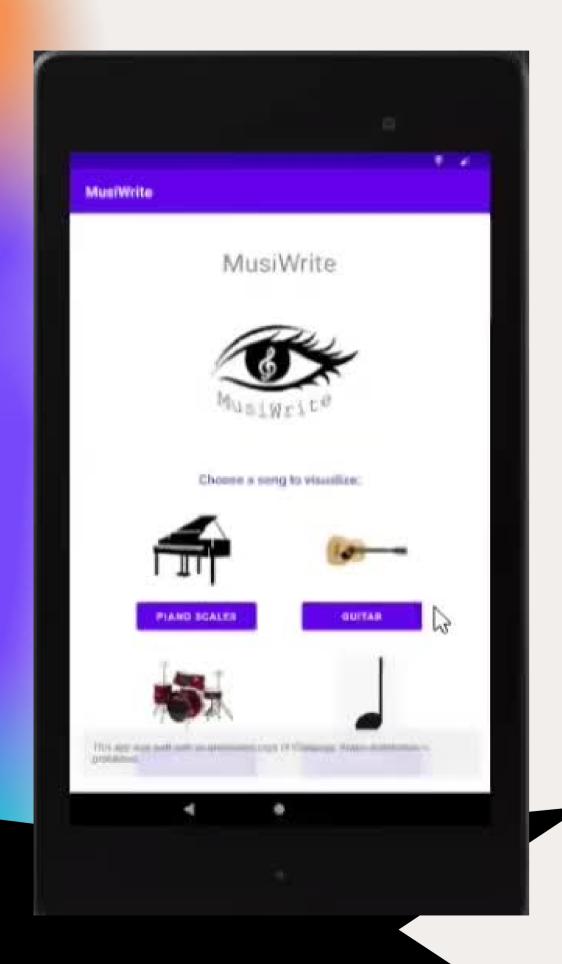
The App visualizes the song in several colorful bars according to Rhythm, Frequency, and Pitch



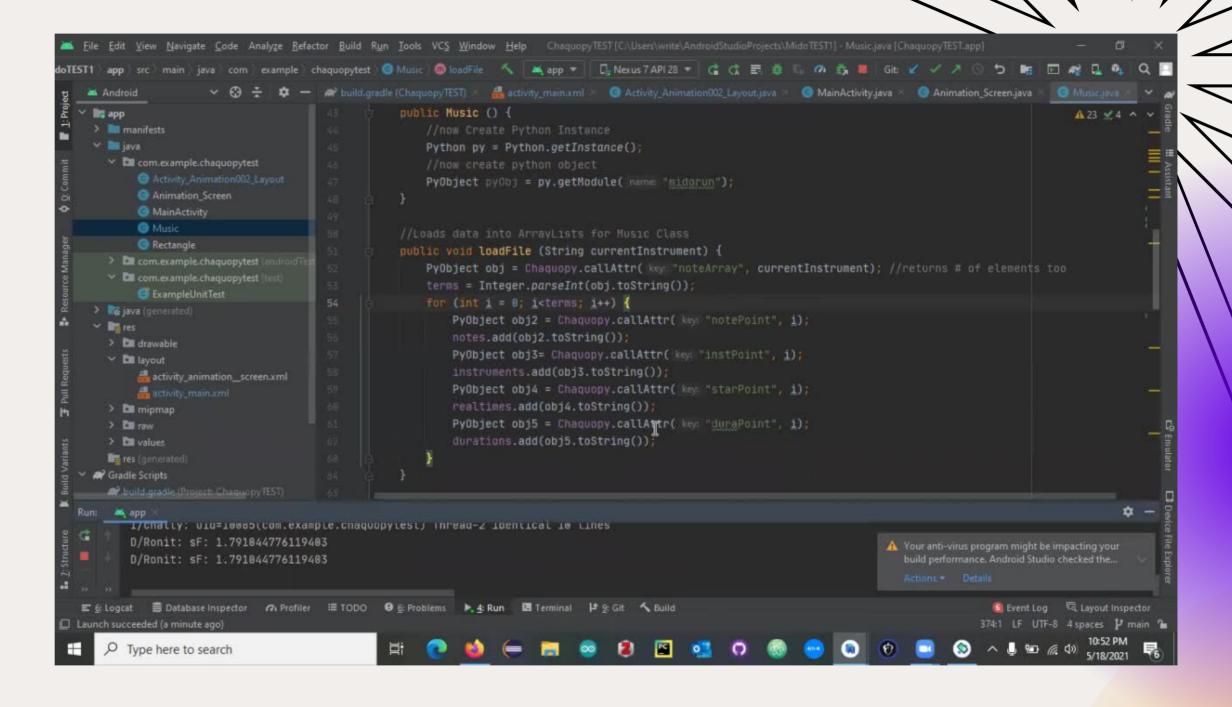
Playback

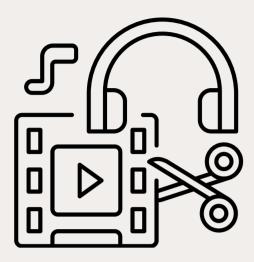
The App can play the music along with the bars, with included features of scrolling and pausing

MUSIWRITE | MASS ACADEMY



App Demo





Tools & Technologies



Python MIDO Library

This library allows the app to read MIDI music files to obtain data such as the pitch, start time, duration, and instruments for each note



Android Canvas Class

This class was used to draw rectangles and animate them based the data obtained



Chaquopy Enviroment

This tool was used to integrate the Python and Java aspects together

Algorithms

Music Analyzation <

- Python Executable builds lists of notes and attributes (4 lists: notes, start times, durations, and instrument)
- 2 MIDO Python Library to read MIDI files
- Chaquopy | Allows Java to run commands in Python
- 4 Music Class in Java allows for getting and setting 4 ArrayLists

Animation



- Uses Chaquopy to obtain music data stored into 4 ArrayLists
- X and Y values for each Note Rectangle are calculated
- Rectangles are drawn on Canvas
- Animated by refreshing the Canvas every 50 ms and translating rectangles 5 pixels
- **5** MP3 audio plays in parallel



PYTHON

- Reads MIDI file
- Sends array of notes

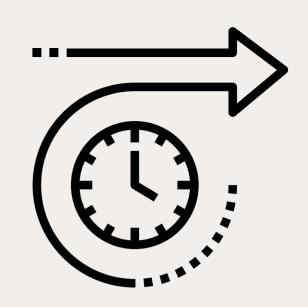
MUSIC

- Stores notes
- Controls animation

ANIMATION

- Canvas
- Sound synched

Future Extensions



ADD ABILITY TO
SEARCH FOR MUSIC



- a feature common in popular music player apps but missing in audio visualizer apps
- would allow the user to choose music outside of the provided local files

- user would not be required to search up their intended song each time
- increases efficiency through easier access of the songs and decreases loading time by saving song data

Any Questions?