JustPerform

“A better karaoke experience”

Team D1
Aleks Watkins, Hugo Martinez, Kiera Cawley

The Problem

Karaoke at-home experience is fragmented. Users must secure equipment, search different platforms to look for songs that may have suboptimal audio, and do not receive feedback.

Our Solution

JustPerform creates one seamless karaoke experience that removes audio from users' personal music library and analyzes audio and motion to provide feedback for improvement.

Areas: Software Systems, Signals and Systems

Use Case Requirements

Working Microphone

10ms or less delay

A fully functional microphone will capture the user’s vocals and send the data to the speaker and the computer. Additionally, the microphone will be equipped with an accelerometer and send the user’s motion data to the computer.

Working Speaker

10ms or less delay

A fully functional speaker will output the vocals of the user combined with the instrumentals of the chosen track.

Use Case Requirements (cont.)

Song Accuracy

80% vocal removal
98% accurate lyrics

Our project should be capable of accurately removing vocals and providing the lyrics of the chosen song to the user to provide the fundamentals of the karaoke experience.

User Feedback System

90% user agreement in testing

Our project should be capable of giving the user a rating based on their skills. We plan to take into account how accurate their pitch and tempo are as well as how closely their movements match up to the beat.

Low Latency

<10s to start song
<10s to release rating

Our project should be capable of starting both the karaoke and rating experience in a reasonable timeframe to avoid friction.

Technical Challenges

Audio Processing:

- Retrieving a clear enough signal from both song and microphone to detect accurate pitch.

Potential Solution:

- We plan to filter out noise from the microphone signal and source clear song audios from Spotify. We will focus on songs with one primary singer.

Technical Challenges

Working in real time/low latency environment:

- Retrieving lyrics, filtering out song vocals, and audio/motion processing should be done, if not in real-time, in very low latency to protect user experience.

Potential Solution:

- We plan to first try using a hardware or software filter for the separation of the instrumentals from the song to avoid the computational delay that may come with more involved software audio removal solutions (e.g. Audacity).
- Cacheing previously loaded song data

Technical Challenges

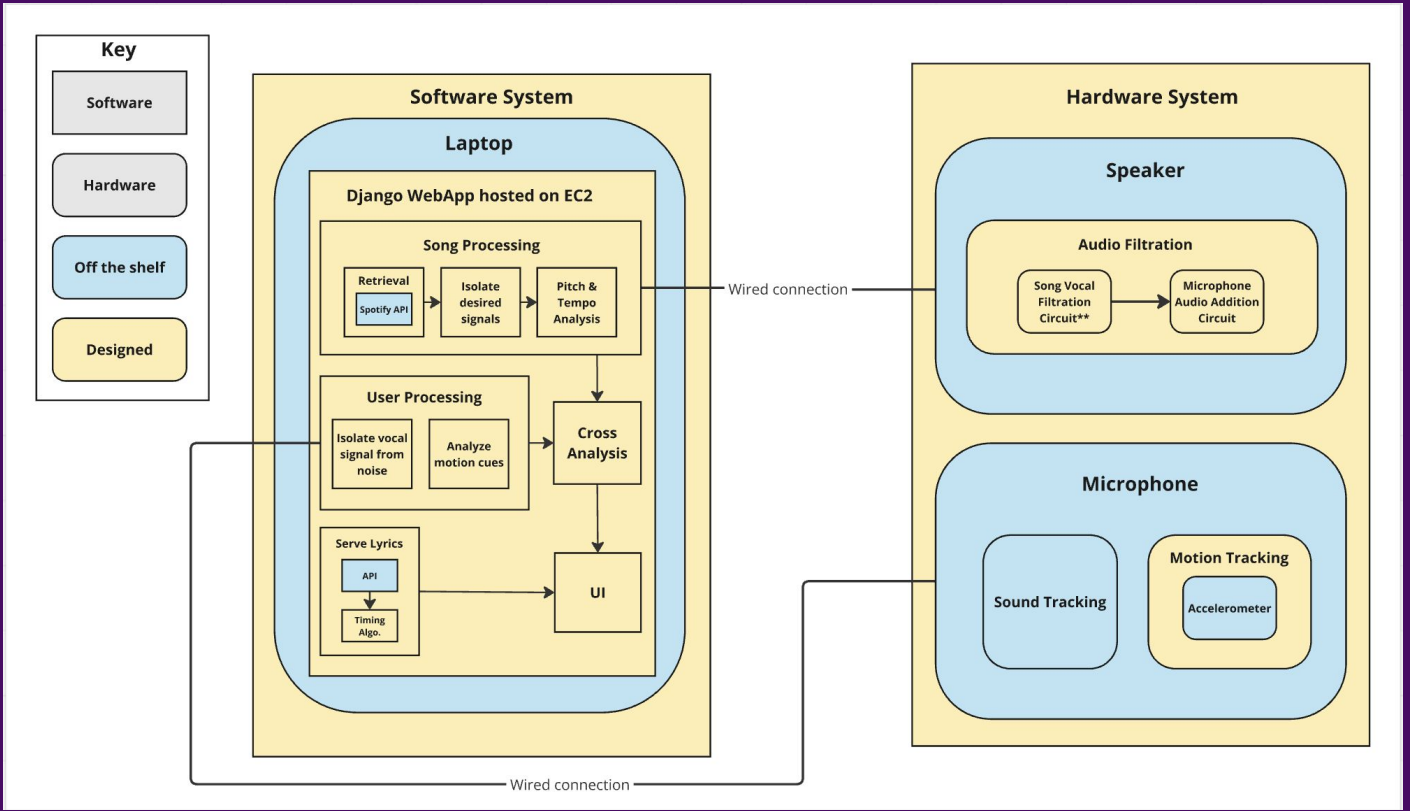
Lyrics:

- Retrieval of not only song lyrics, but lyric timing for accurate display given the leading tech (MusixMatch) is not taking educational use applications.

Potential Solutions:

- Scrape Genius.com or a similar lyrics website (may be challenging to retrieve timestamps)
- Explore reliability of existing APIs
- Use voice to text software to generate the lyrics from the audio (need to process the song before the user can start)

Solution Approach



Testing Verification and Metrics

Singer vocals
90% Accurate

The singer's pitch will be noted with a tuner, and then later compared to the pitch identified by the software from our extracted vocals.

Singer movement
90% Accurate

The microphone will be moved according to a series of simple movements (direction and velocity changes) to check accurate decoding of sensor readings. Then the microphone will move along with a metronome to test our softwares ability to detect tempo.

Music Library
80% Accurate

We will play different genres of music through our speaker in order to assess the quality of vocal removal. Additionally, putting the output track through our vocal signal isolator can identify remaining artifacts. The lyrics we get for these songs will also be compared to trusted online sources and validated.

Tasks & Division of Labor

Kiera
Hardware Lead

- Microphone motion tracking
- Integration of components
- Lyric sourcing

Hugo
Audio Lead

- Song vocal filtration
- Isolate vocal signals lead*
- Pitch analysis lead*

Aleks
Software Lead

- Web Application set up & Deployment
- Playable music
  - Spotify API
- Frontend displays
- Lyric Sourcing

*Audio analysis planned as most collaborative component due to perceived difficulty.

GANTT Chart

	Week 0	Week 0.5	Week 1	Week 1.5	Week 2	Week 2.5	Week 3	Week 3.5	Week 4	Week 4.5	Week 5
Kiera	Accelerometer set up	Accelerometer Connection	Analyze Basic Movement	Source lyrics and timing information	Slack / Collaboratively finetuning and testing audio processing	Analyze movement for tempo	Testing movement tracking	Full integration and testing	Movement finetuning	Slack	Slack
Aleks	Spotify API: ensure playable music	Figma UI Design	Set up basic WebApp Capabilities			Lyric display in-time karoake style UI	Callabrating scoring system		Deployment and UI Finetuning	Slack	Slack
Hugo	Pull singer signal from microphone	Analyze singer signal for basic pitch	Isolate vocal signal from music	Analyze song signal for pitch		Removing singer vocals from track	Testing vocal removal		Audio finetuning	Slack	Slack