Abstract

## Project Abstract

Immerse yourself in the world of music with SightStep, a cutting-edge computer vision application. Transform any space into your private dance floor and karaoke stage. SightStep listens to your favorite karaoke songs, tracks your movements, and guides you through dance routines. Sing along effortlessly as the lyrics sync with your performance. Elevate your karaoke experience with SightStep's fusion of computer vision, music, and dance. Get ready to step into a new realm of entertainment!

## User Stories and Design Diagrams

### User Stories

- As a casual singer, I want to visit the website, pick a song, and use only video scoring to have a lighthearted and entertaining karaoke experience without worrying about my singing skills.

- As a song enthusiast, I want to access the app, use both video and audio scoring, and choose from a wide range of songs from the database to challenge myself and have fun with friends.

- As a shy performer, I want to use the app in audio scoring mode, follow along with the lyrics displayed, and improve my singing skills privately while enjoying the music.

### Design Diagrams

#### Level 0 Diagram

<img width="562" alt="Screenshot 2023-11-22 at 8 46 44 PM" src="https://github.com/mullise147/VisionVanguards/assets/68687725/565757fc-f9cd-46d1-a1b7-cdefaf0d3fe9">

#### Level 1 Diagram

<img width="1043" alt="Screenshot 2023-11-22 at 8 46 52 PM" src="https://github.com/mullise147/VisionVanguards/assets/68687725/7928acf6-8ca5-4a0c-ba39-f634eba0aa41">

#### Level 2 Diagram

<img width="608" alt="Screenshot 2023-11-22 at 8 47 02 PM" src="https://github.com/mullise147/VisionVanguards/assets/68687725/b7495085-877d-425c-862a-1da97869c50c">

#### Description of Diagrams

As we go from level 0 to level 2, the level of detail starts to heighten.

Level 1: A user can start a session, and perform a song and hit the "replay" button to replay the song and if not, end the singing session.

Level 2: A user can start a session, and request mic and camera access and then begin to select a song from the list. Once enabled, the user performs the song (if only mic then audio, if both then video). Then the points are calculated and shown to the user, after which the user is able to restart a session.

Level 3: This talks about the leaderboard aspect of the application. Once a score is displayed, a user can choose to add it to the leaderboard or not and make the score public.

## Project Tasks and Timeline

### Task List

All developers will work mainly on their section but will also do some work on the other sections.

Athulya Ganesh - Lead web interface developer.

Stephanie Mullins - Lead Computer Vision developer.

Rob Kelly - Lead audio processing developer.

Computer Vision Tasks (Stephanie):

- Research technologies (gesture recognition, skeleton detection, skeleton tracking w/ comparison to given position) and how others have done problems like our project with a focus on gesture recognition and tracking.

- Implement gesture recognition for basic hand signals (picture then live video).

- Expand gesture recognition to include gesture tracking of hand signals (recorded then live video).

- Research technologies for skeleton recognition and tracking and see how others have implemented the problem.

- Implement skeleton detection to see if a human can get recognized as a skeletal figure (picture then live video).

- Expand skeleton detection to include skeletal tracking to convert the whole human body to a skeletal figure (recorded then live video).

- Implement skeletal tracking correctness compared to given positions to calculate graded score.

Audio Processing Tasks (Rob):

- Research various audio processing libraries (as well as experiment with them) in order to find the best library for the needs of the project Some libraries to consider include Speech Recognition, Librosa, IPython, TorchAudio, PyAudio

- After selecting the best library for the project, implement it within the team project

- In particular, implement audio processing tool so that it can be manipulated in various ways that will be necessary for the project

- Research methods of implementing audio to text transcription

- Research various methods to implement pitch detection within the project (used to determine the accuracy of the pitch of the user compared to the audio file of the song)

- Implement both of the above features to be able to determine a user’s lyrical and pitch accuracy

- Using the above features, implement algorithm to compute composite accuracy of a user’s singing on a given song

Web Interface Tasks (Athulya):

- Research and select appropriate frontend and backend technologies for the web app, as well as database options (consider both local and cloud services for scalability) and suitable APIs for feature integration

- Develop page-by-page wireframes and prototypes using Figma, including the landing page, performance options, lyrics display, gesture list, scoring interface, and leaderboard

- Set up the development environment, including configuring the chosen frontend and backend frameworks

- Prepare the server and database environment, ensuring proper access and security settings

- Develop the initial skeleton of the web app with basic navigation and structure, including landing page for user selections and different interfaces for audio-only and audio-video performances

- Implement performance recording and playback functionality

- Develop the module to display lyrics on the screen in sync with the chosen song, ensuring that the GUI is user-friendly

- Create a module that lists gestures for users to follow along during performances

- Implement features to highlight or track user gestures in real-time

- Develop a scoring system to evaluate user performances, as well as a leaderboard for users to view their performance in relation to others

- Develop user registration and login pages to save user information securely

- Set up the backend database, either locally or using a cloud service, to store user profiles, scores, and song information

- Perform comprehensive testing at each development stage, including unit testing, integration testing, and user testing

- Deploy the web application within the specified timeline using a hosting service like Vercel or Netlify

Combined Tasks:

- Hook package containing all of the audio processing logic into the project web application

- Combine computer vision aspects with audio and website portions

- Implement an algorithm to calculate composite scores of audio and visual aspects of performance

### Timeline

By the end of this, we expect the following.

<img width="599" alt="Screenshot 2023-11-22 at 9 05 02 PM" src="https://github.com/mullise147/VisionVanguards/assets/68687725/ada5428c-8f49-4231-a3cd-70a655588f1a">

### Effort Matrix

<img width="387" alt="Screenshot 2023-11-22 at 9 05 42 PM" src="https://github.com/mullise147/VisionVanguards/assets/68687725/85520f97-73ec-498c-980c-d640831fb1fd">

<img width="385" alt="Screenshot 2023-11-22 at 9 05 53 PM" src="https://github.com/mullise147/VisionVanguards/assets/68687725/e92b8d1e-cbb6-41da-bd26-db1dd40a6f95">

<img width="386" alt="Screenshot 2023-11-22 at 9 06 00 PM" src="https://github.com/mullise147/VisionVanguards/assets/68687725/48afabc8-c701-4f5c-84ed-2e99fcdfee3d">

<img width="384" alt="Screenshot 2023-11-22 at 9 06 06 PM" src="https://github.com/mullise147/VisionVanguards/assets/68687725/f29a43d8-8295-4a06-a7a6-3ea172ae7309">

<img width="381" alt="Screenshot 2023-11-22 at 9 06 11 PM" src="https://github.com/mullise147/VisionVanguards/assets/68687725/82e212f7-74b7-4fe6-b4f1-31edec53ea43">

## ABET Concerns Essay

View here: <https://docs.google.com/document/d/1UlMg2LQ5EyT-aBHIn1_bGkr_9jN04lpPhCgV6xu_uHY/edit?usp=sharing>

## Budget

- Expenses to Date: $0

- Donated Items: $0