EMOTION BASED MUSIC PLAYER

Dharmsinh Desai University
Nadiad

Developed By:
IT093 - Patel Dirgh
IT108 - Raycha Pranay

Internal Guide:
Prof. Archana Vyas
Dept. of Information Technology
FOT DD University

EMOTION BASED MUSIC PLAYER

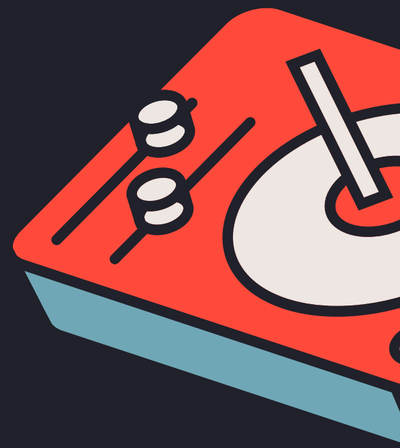


TABLE OF CONTENT

INTRODUCTION

- Facial expressions are a great indicator of the state of the mind for a person. Humans tend to link the music they listen to, to the emotion they are feeling. The song playlists though are, at times too large to sort out automatically. It would be helpful if the music player was “smart enough” to sort out the music based on the current state of emotion the person is feeling.

- The application is developed in such a way that it can analyze the image properties and determine the mood of the user.

PURPOSE

- The main purpose of this system is to provide users a better and faster way to suggest music. We aim to build an application that focuses on reducing human efforts by generating a playlist based on facial sentiments.

SCOPE

- Our application can be used by the users who have downloaded and installed our .exe file on their local disk.

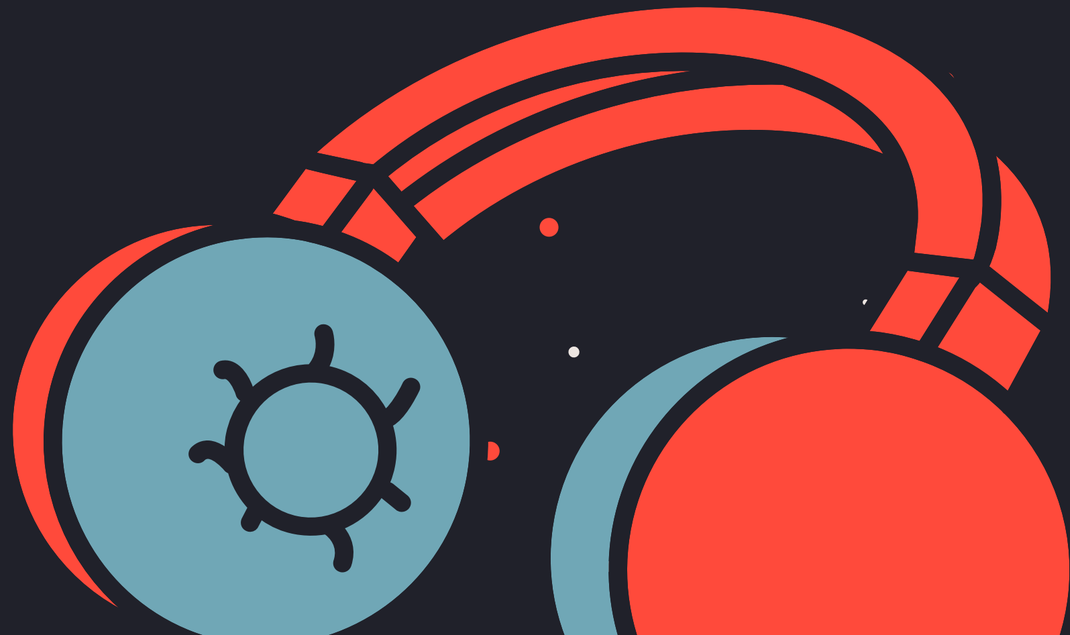
REQUIREMENTS FOR APPLICATION

1. Software Requirements:

- Python version 2.7 or above
- OpenCV
- Keras, Tensorflow, numpy

2. Hardware Requirements:

- Microsoft® Windows® 7/8/10.
- 2 GB RAM minimum, 8 GB RAM recommended
- Web-cam.



REQUIREMENTS OF SYSTEM (SRS)

R1: Camera permission phase

Description: Users have to allow the system to use the web-cam to capture images.
Input: Click “Yes” or “No”.
Output: If users click on the system will exit else continue to capture the image.

R2: Emotion Capture phase

Description: As soon as the permission phase is done, the application will capture the user’s image through a web-cam and start detecting the user's emotion.
Input: User face.
Processing: Scanning of face.
Output: Face detected.

R3: Image Processing Phase

Description: After the image is captured, the application sends the captured image for processing and after the captured image is processed the image feedback is sent to the application.
Input: Scanned face from Emotion Capture Phase.
Output: Image feedback is sent to the application.

REQUIREMENTS OF SYSTEM (SRS)

R4. Emotion Detection Phase

Description: In this phase, the application receives the image information and recognizes the emotion based on the defined threshold. This emotion is then used to fetch the emotion's play-list.
Input: Input from image processing phase.
Processing: Detection of emotion.
Output: Emotion generated.

R5: Confirmation Phase

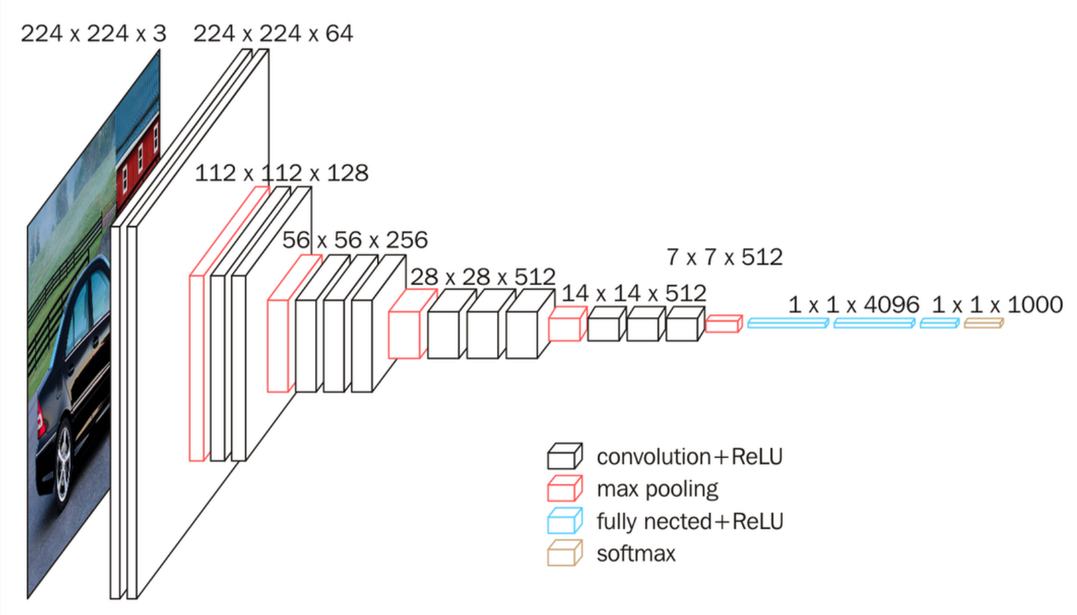
Description: Here the user will be prompted by a pop up confirming the detected emotion. If the user is not satisfied with the emotion detected, they can re-detect it.
Input: Confirmation pop-up: "Confirm Mood"
Processing: If "Yes" the system starts playing song, if "No" re-scan face.

R6: Display phase

Description: Here the songs are provided to the user and the user can play any song from the list displayed. The user has the option to play, pause, skip forward, skip backward and also pause a song in the application.

TECHNOLOGY AND LITERATURE REVIEW

- We started by classifying emotions into five major emotions, (Happy, Sad, Angry, Surprise, and Neutral) and also using Convolutional Neural Network and we got over 72% as accuracy.



Architecture a CNN model

- Our CNN model has multiple layers to classify mood.

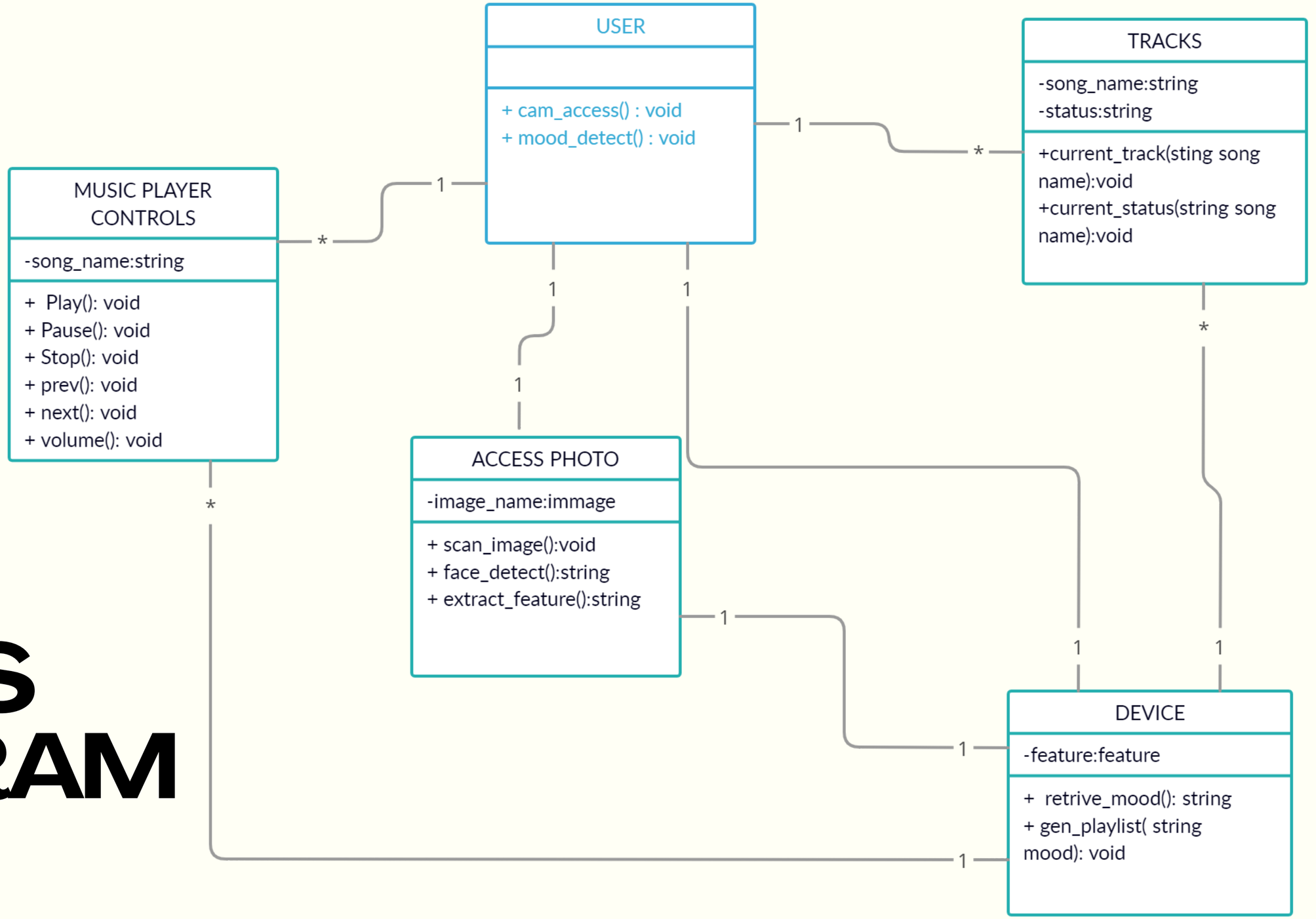
- The dataset is (48 x 48) pixels grayscale images. The number of images in this dataset that's been used (24282 image), (7164 for Happy, 4938 for Sad, 4982 for Neutral, and 3993 for angry, 3205 for surprise).

- We have used Tkinter which is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications.

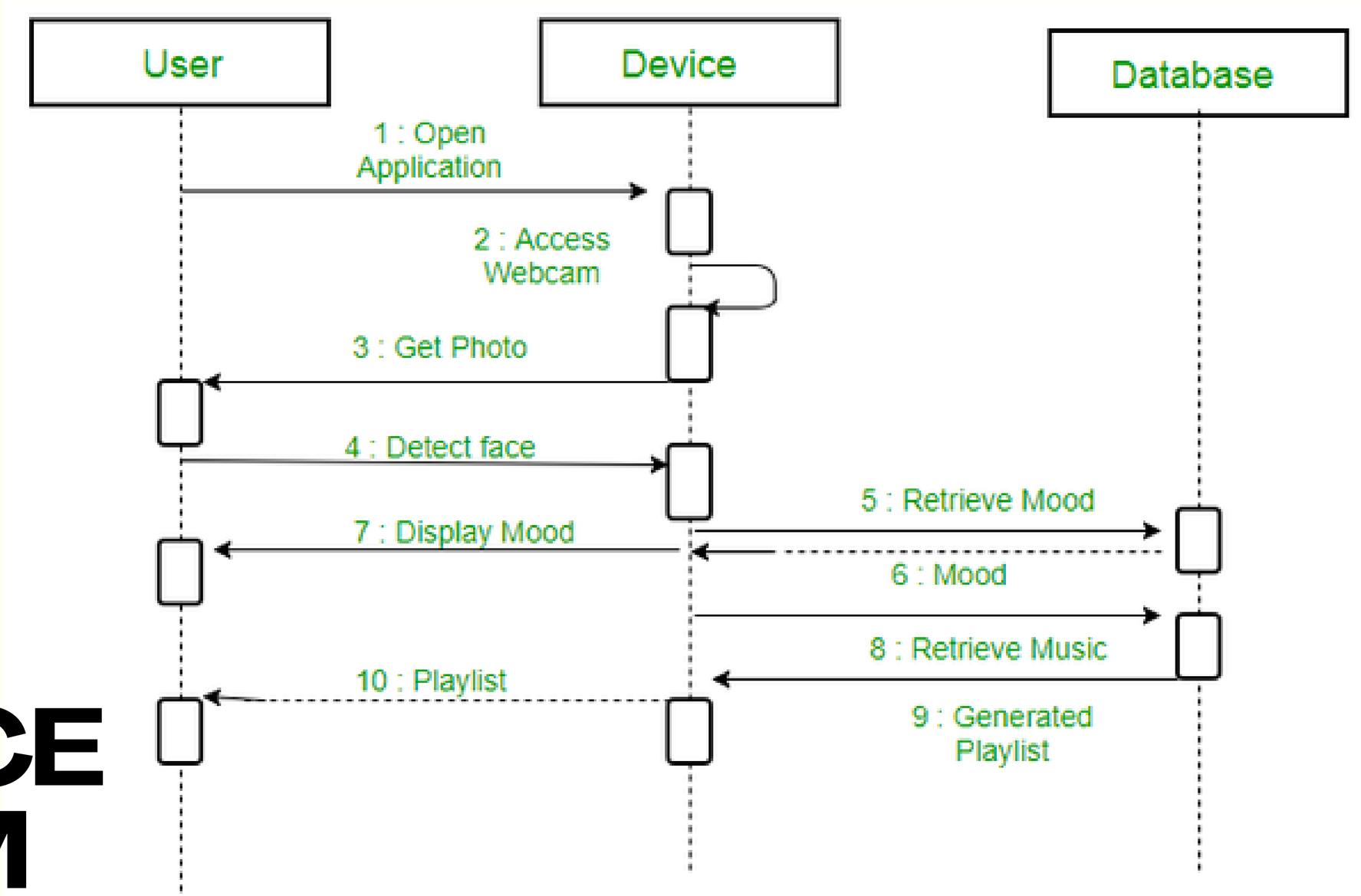
UML DIAGRAMS

USE CASE DIAGRAM

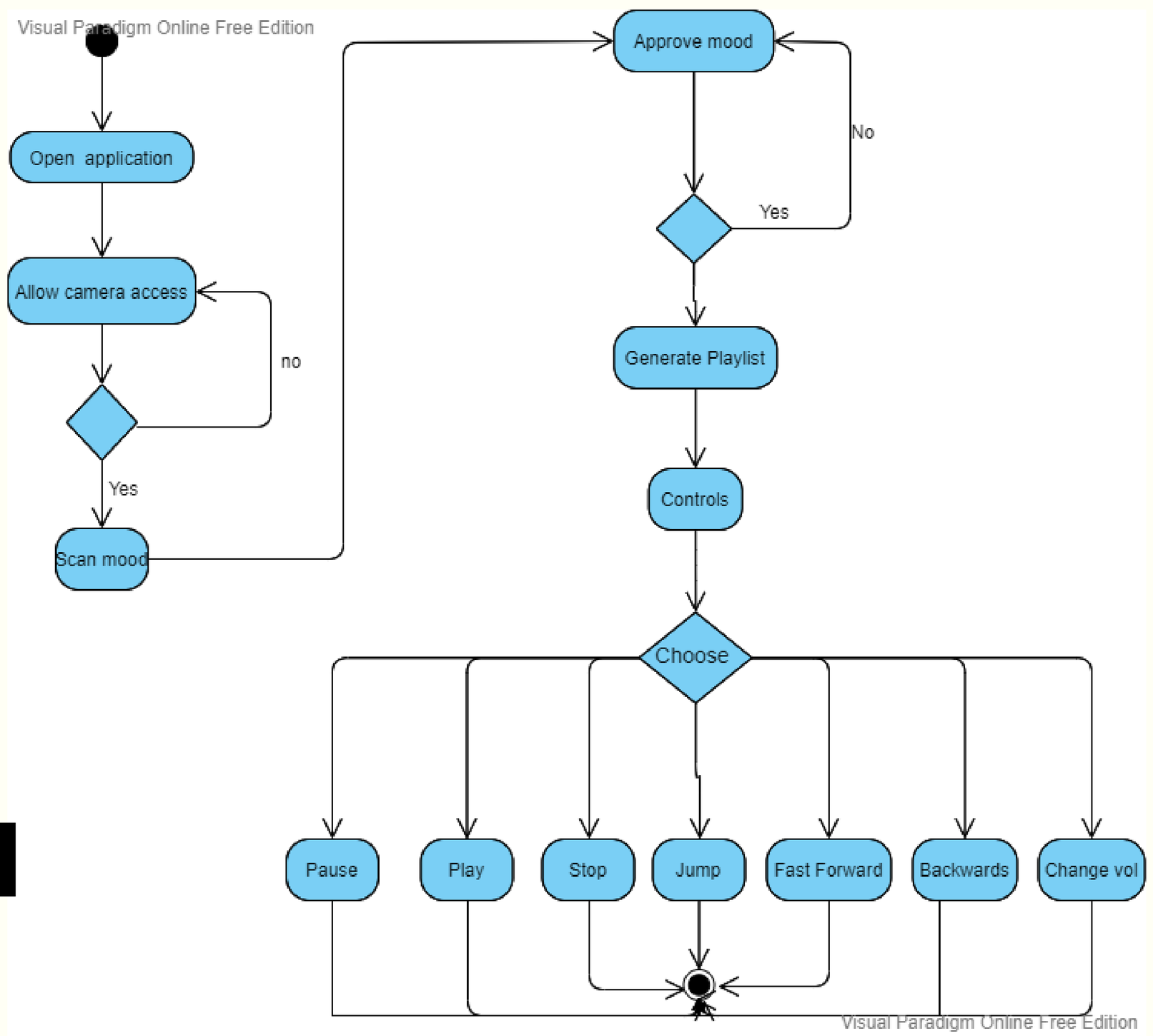
CLASS DIAGRAM



SEQUENCE DIAGRAM



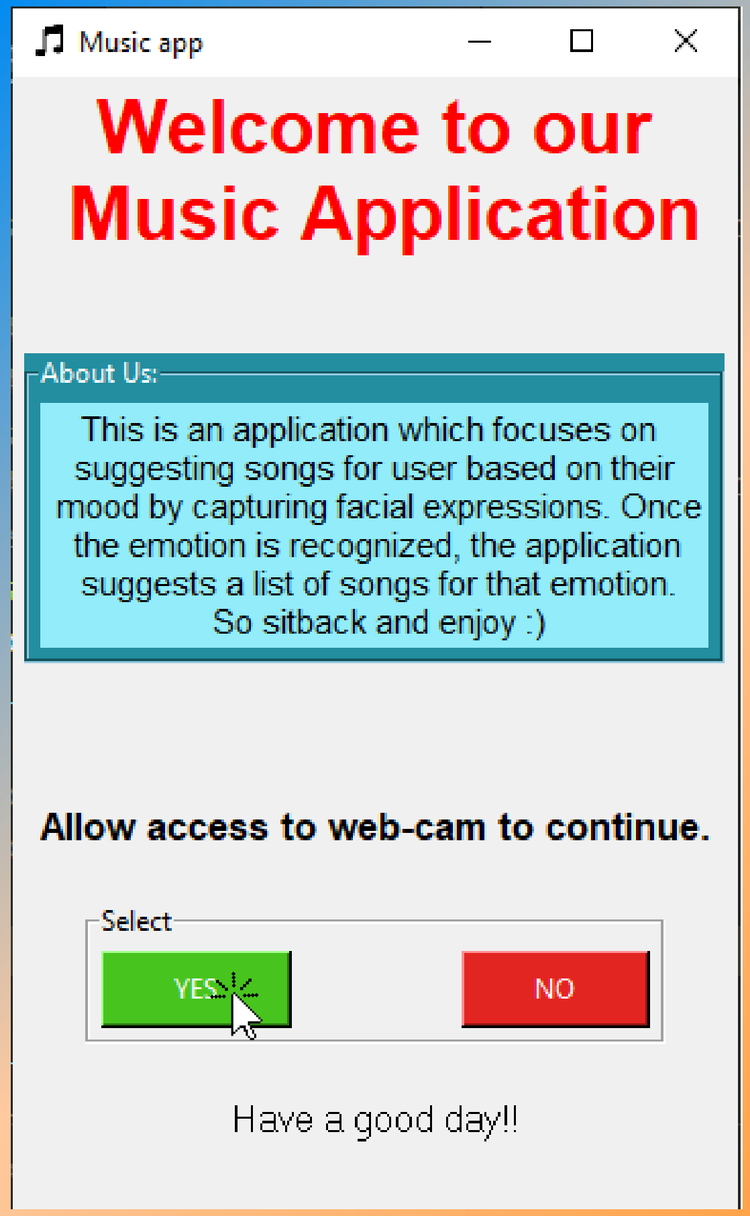
ACTIVITY DIAGRAM



SCREEN SHOTS

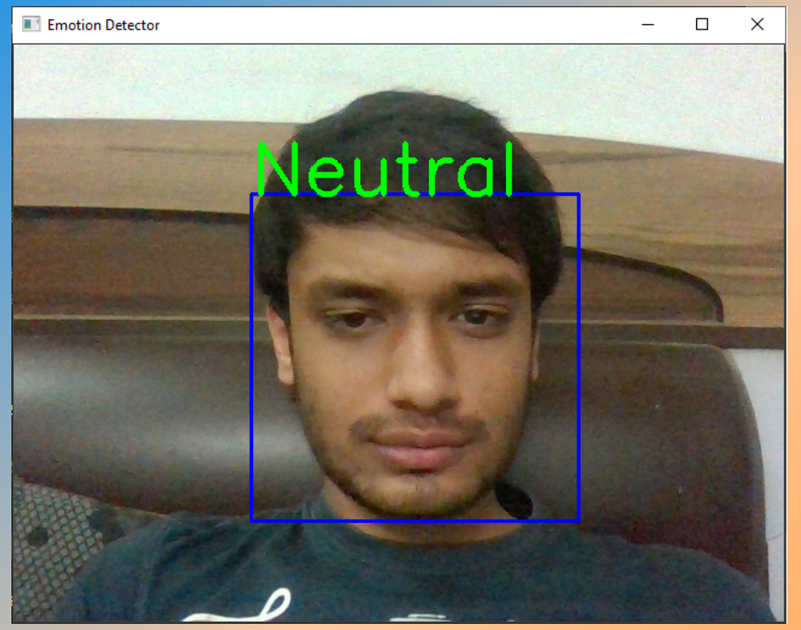
HOME PAGE

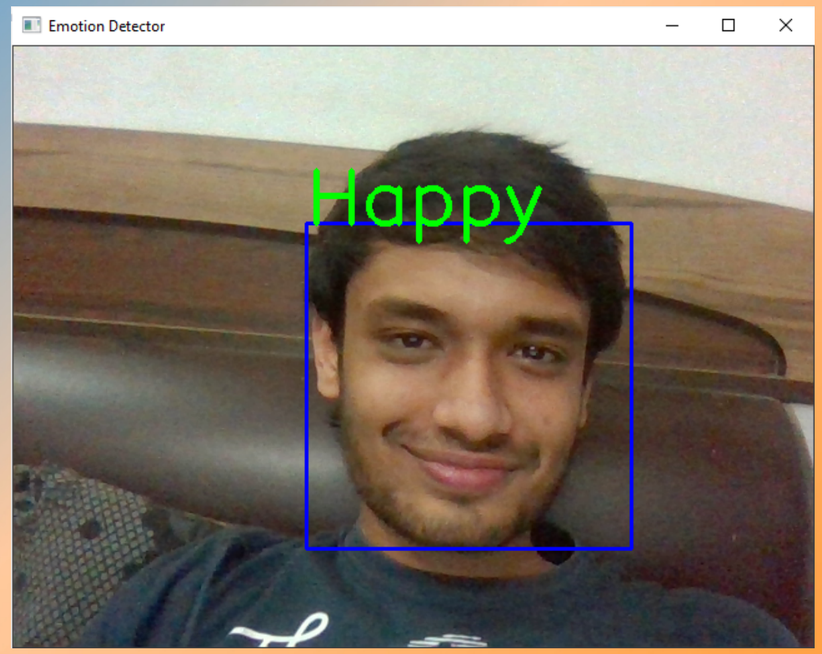
CLICK 'YES' TO GET STARTED
CLICK 'NO' TO QUIT



EMOTION DETECTION

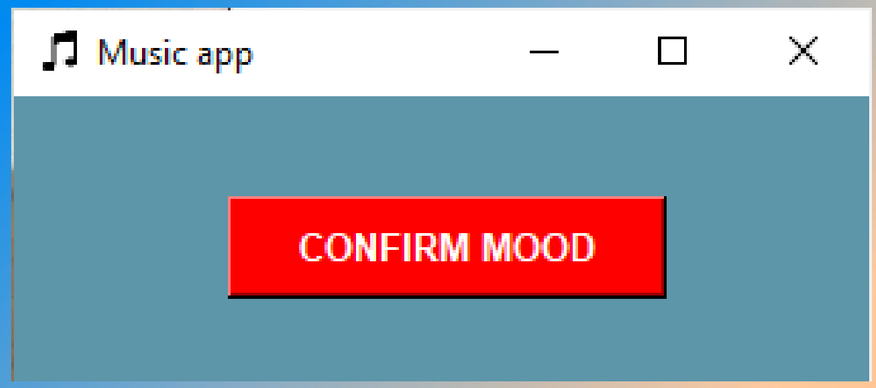
CLICK 'Q' TO CLOSE WINDOW

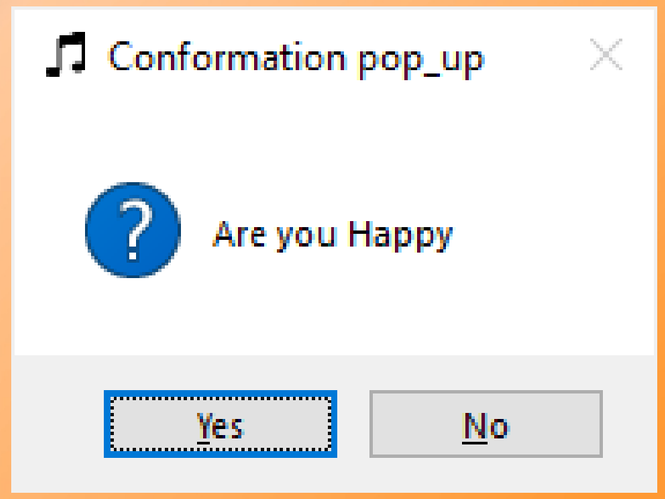




CONFIRM EMOTION

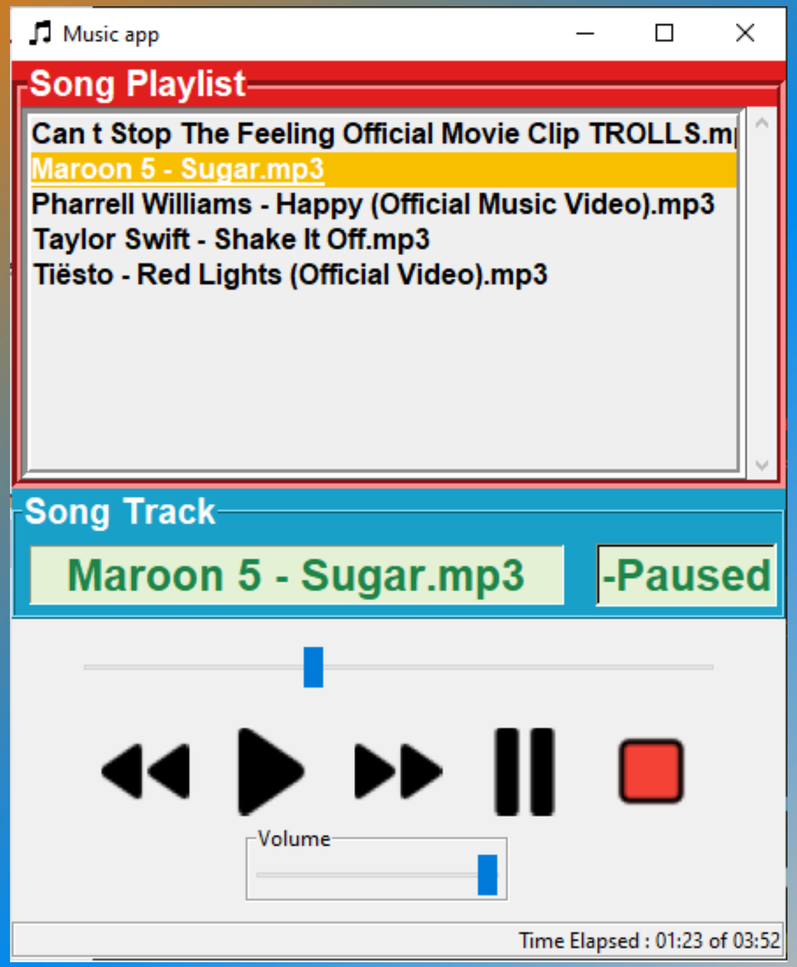
CLICK 'YES' TO CONTINUE
CLICK 'NO' TO RECHECK EMOTION

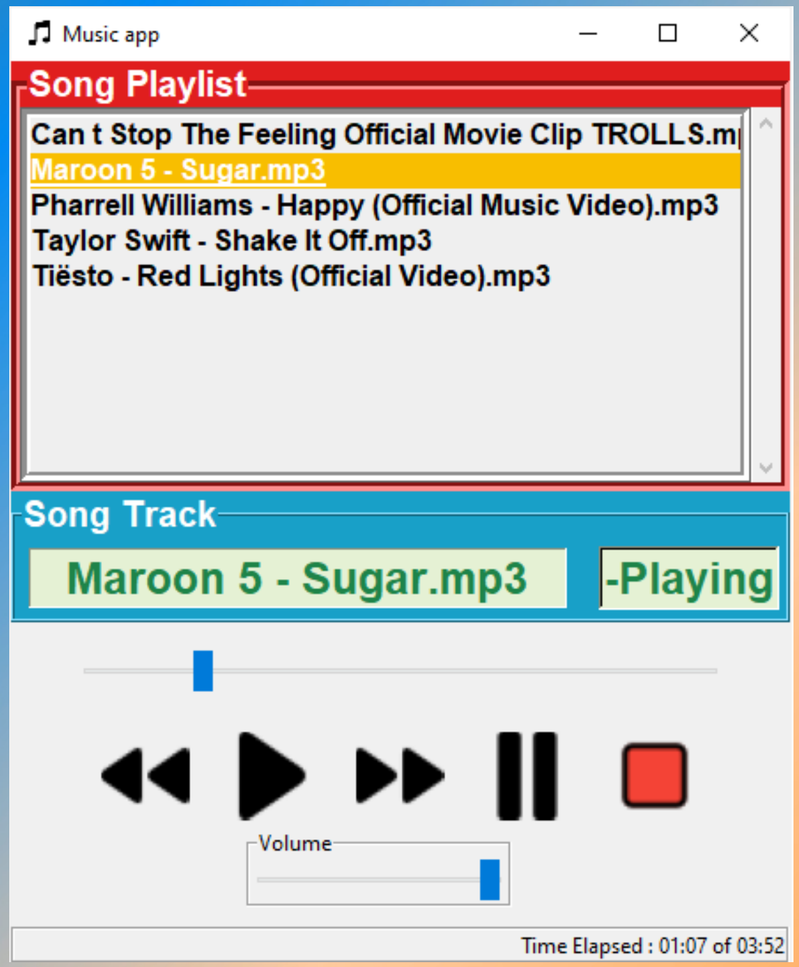


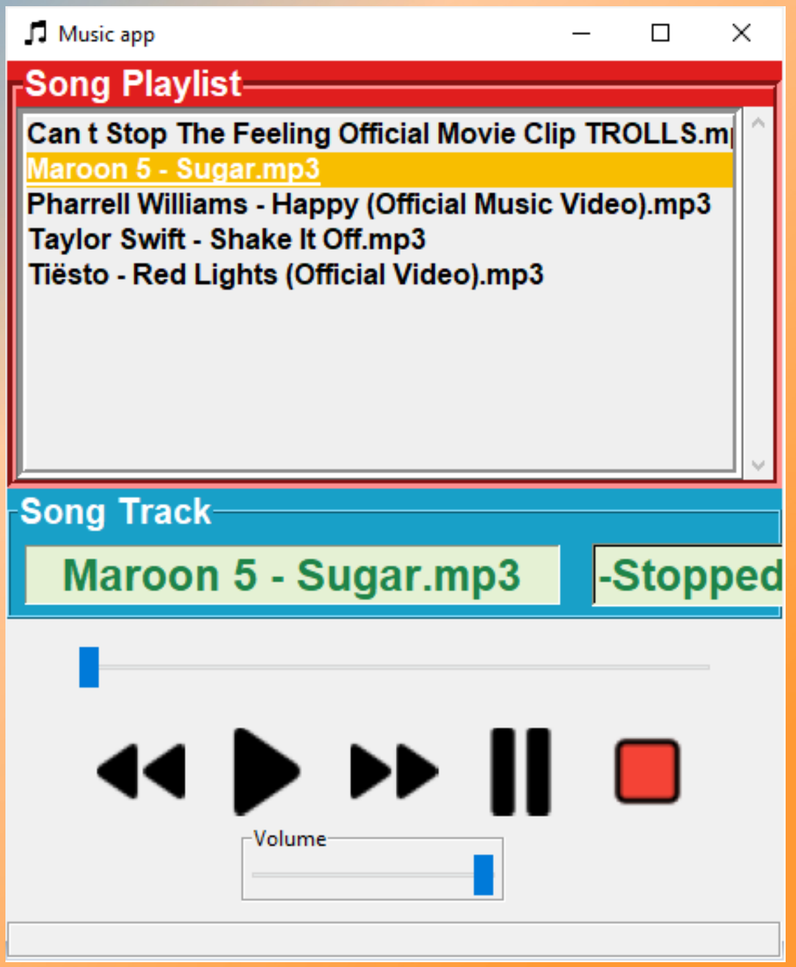


MUSIC PLAYER

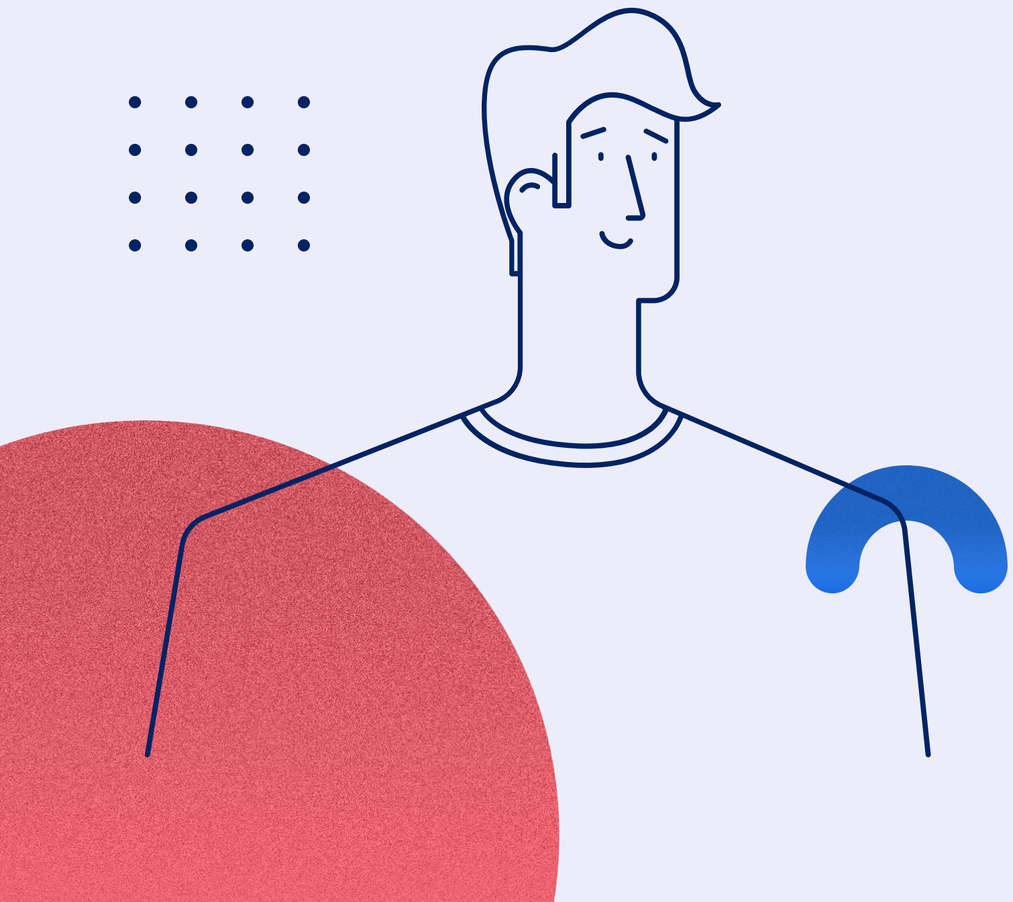
WITH VARIOUS CONTROLS







CHALLENGES FACED



- First problem we faced was to gain good accuracy , in our project we used CNN technique and first we kept no of epochs 25 ,that time we were getting 60 % accuracy.

- In the GUI part we faced many problems like, if we paused and then unpaused any song the song would restart instead of continuing from where we paused it.

- While importing one file into another file just for the use of the emotion which was detected we discovered that the system went into a loop of detecting emotion.

- At low light and low resolution our app is not so accurate, we are working on that.

FUTURE EXPANSION

- User can manually add their own playlist to the application. And with the help of reading the metadata of an MP3 the songs can be segregated accordingly.

- Instead of fetching the songs from a local device, one can stream songs directly from the cloud.

- The proposed system also tends to avoid the unpredictable results produced in extreme bad light conditions and very poor camera resolution.

REFERENCES

- Reference docs:
  - For GUI :
    https://www.geeksforgeeks.org/python-tkinter-tutorial/
  - For OpenCV:
    https://www.tutorialspoint.com/opencv/index.htm
    https://youtube.com/playlist?list=PLQVvvaa0QuDdttJXlLtAJxJetJcqmqlQq
  - For Deeping Learning:
    https://youtube.com/playlist?list=PLZoTAELRMXVPkl7oRvzyNnyj1HS4wt2K-
- Reference sites for debugging:
  Stackoverflow, Reddit, GitHub.

Thank you!