

***SENTIMENT ANALYSIS WITH EMOTION BASED  
AUDIO PLAYER***

***A Project report***

**Submitted in partial fulfillment of the requirements  
for the award of the**

**BACHELOR OF TECHNOLOGY  
IN  
INFORMATION TECHNOLOGY**

**Submitted By**

***N. SRILEKHA -17BQ1A12B2  
K. NESHMA -17BQ1A1266  
L. KAVYA SRI -17BQ1A1291***

**Under the supervision of**

***Mrs. G. SIRISHA***

***Assistant Professor, Dept of IT, VVIT***



**VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY**

NAMBUR (V), PEDAKAKANI (M), GUNTUR-522 508, TEL no: 0873 2118036,

url: [www.vvitguntur.com](http://www.vvitguntur.com). Approved by AICTE, permanently affiliated to JNTUK  
Accredited by NAAC with “A” grade, Accredited by NBA for 3 years.

***July 2021***

# **VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY NAMBUR**



## **BONAFIDE CERTIFICATE**

This is to certify that this project report is the bonafied work of “**N. SRILEKHA (17BQ1A12B2), K. NESHMA (17BQ1A1266), L. KAVYA SRI (17BQ1A1291)**”, who carried out the project under my guidance during the academic year 2020-2021 towards partial fulfillment of the requirements of the Degree of **Bachelor of Technology in Information Technology** from Jawaharlal Nehru Technological University, Kakinada. The results embodied in this report have not been submitted to any other University for the award of any degree.

### **PROJECT GUIDE**

**Mrs.G.SIRISHA**  
**Assistant Professor**

### **HEAD OF THE DEPARTMENT**

**Dr. KALAVATHI ALLA, Ph.D**  
**Professor**

**Submitted Viva voce Examination held on:** \_\_\_\_\_

**External Examiner**

## **CERTIFICATE OF AUTHENTICATION**

We, Ms. N.Srilekha, Ms. K.Neshma , Ms. L.Kavya Sri hereby declare that the Project Report entitled “**SENTIMENT ANALYSIS WITH EMOTION BASED MUSIC PLAYER**” done by us under the guidance of **Mrs. G.Sirisha**, Assistant Professor, Information Technology at Vasireddy Venkatadri Institute of Technology is submitted for partial fulfillment of the requirements for the award of Bachelor of Technology in Information Technology. The results embodied in this report have NOT been submitted to any other University for the award of any degree. It is further assured that this work has NOT been submitted, either in part or in full, to any other department of the Jawaharlal Nehru Technological University, or any other University, Institution or elsewhere, except in conference or journal publication authored by us.

DATE :

PLACE :

**SIGNATURES**

**NAIDU SRILEKHA - 17BQ1A12B2**

**KASAMNENI NESHMA-17BQ1A1266**

**LINGINENI KAVYA SRI- 17BQ1A1291**

## ACKNOWLEDGEMENT

We take this opportunity to express our deepest gratitude and appreciation to all those people who made this project work easier with words of encouragement, motivation, discipline, and faith by offering different places to look to expand my ideas and help us towards the successful completion of this project work.

First and foremost, we express our deep gratitude to **Mr. Vasireddy Vidyasagar, Chairman**, Vasireddy Venkatadri Institute of Technology for providing necessary facilities throughout the Information Technology program.

We express our sincere thanks to **Dr. Y. Mallikarjuna Reddy, Principal**, Vasireddy Venkatadri Institute of Technology for his constant support and cooperation throughout the Information Technology program.

We express our sincere gratitude to **Dr.A.Kalavathi, Professor & HOD**, Information Technology, Vasireddy Venkatadri Institute of Technology for her constant encouragement, motivation and faith by offering different places to look to expand my ideas. We would like to express our sincere gratitude to our guide **Mrs.G.Sirisha , Assistant Professor** and Project coordinator **Dr.P.L.Kishan Kumar Reddy, Associate Professor**, Department of Information Technology for their insightful advice, motivating suggestions, invaluable guidance, help and support in successful completion of this project.

We would like to take this opportunity to express our thanks to the **Teaching and Non-Teaching staff** in the Department of Information Technology, VVIT for their invaluable help and support.

N.Srilekha

K.Neshma

L.Kavya Sri

# **SENTIMENT ANALYSIS WITH EMOTION BASED AUDIO PLAYER**

## **ABSTRACT**

People experience stress due to various factors like work pressure, emotional problems, disaster, violence, etc in their day to day life. This stress leads to many physical and mental health risks such as Asthma, Headaches, Anxiety, heart disease, Depression, Asthma, Alzheimer's disease, etc. Music therapy has the ability to balance both the physical and mental fitness of humans. Music therapy is a healing process that uses music to inscribe the emotional, physical, cognitive needs of a self one or a group. Manually segregating the list of songs and generating an appropriate playlist based on an individual's emotional feature is very tedious. So here we come up with the system, where the system will play songs based on the expression or mood of the user. This proposed system based on the facial expression extracted will generate a playlist automatically thereby reducing the effort and time involved in rendering the process manually. In this system we can find different types of song playing modes apart from emotion based they are Random mode and queue mode. Here we extract an individual's behavior and emotional state of a person. Facial expressions are captured using an inbuilt camera and are categorized into four types: happy, angry, sad, neutral based on one of these types, songs will be played. Thus, it yields better accuracy in terms of performance and computational time and reduces the designing cost. This system is implemented in Visual studio.

Keywords: Emotions, Music Player, Facial Expression.

## TABLE OF CONTENTS

<b>Title</b>	<b>Pageno</b>
<b>CHAPTER I INTRODUCTION</b>	<b>1</b>
1.1 Introduction	1
1.2 Aim	2
1.3 Scope	3
Goals and Objectives	3
1.4 Existing System	3
1.5 Proposed System	4
1.6 Feasibility Study	6
1.6.1 Economical Feasibility	6
1.6.2 Technical Feasibility	7
1.6.3 Social Feasibility	7
<b>CHAPTER II LITERATURE SURVEY</b>	<b>8</b>
<b>CHAPTER III REQUIREMENTS SPECIFICATION</b>	<b>10</b>
3.1 Description Of Problem	10
3.2 Proposed Solution	10
3.3 Software Requirements	10
3.3.1 Hardware Requirements	10
3.3.2 Software Requirements	11
3.3.3 Software Description	11
3.3.4 Functional Requirements	16
3.3.5 Non-functional Requirements	17
3.3.5.1 Functionality	17
3.3.5.2 Reliability	17
3.3.5.3 Usability	17

3.3.5.4 Safety	17
3.3.5.5 Security	17
3.3.5.6 Robustness	17
3.3.5.7 Communications	18
3.3.6 Modules	18
3.3.6.1 Image Capturing	18
3.3.6.2 Emotion Extraction	18
3.3.6.3 Audio Integration	20
3.3.6.4 Generating Playlist	20
3.4 Uml Diagrams	21
3.5 System Analysis Methods	22
3.5.1 Use Case Diagram	22
3.5.2 Activity Diagram	23
3.6 System Design Methods	24
3.6.1 Class Diagram	24
3.6.2 Sequence Diagram	25
3.7 E-R Diagram	29
<b>CHAPTER IV SYSTEM DESIGN</b>	32
4.1 System Architecture	32
<b>CHAPTER V SYSTEM IMPLEMENTATION</b>	33
5.1 Tools & Techniques	33
5.2 Implementation Code	38
5.3 Screenshots	41
<b>CHAPTER VI SYSTEM TESTING</b>	42
6.1 Purpose of Testing	42
6.2 Testing Strategies	42

6.3 Snapshots Of Test Cases	45
<b>CHAPTER VII CONCLUSION</b>	53
7.1 Conclusion	53
7.2 Future Enhancements	53
<b>CHAPTER VIII Appendices</b>	54
8.1 Sample code Snippets	54
8.2 Bibliography	55



## **LIST OF FIGURES**

Fig 1.1. System Overview	2
Fig 1.2. Existing System Overview	4
Fig 1.3. Proposed System	5
Fig 3.1. Face Feature Extraction	20
Fig 3.2. Integration Of Emotion	21
Fig 3.3. Generating Playlist	21
Fig 3.4. Use Case Diagram	24
Fig 3.5. Activity Diagram	25
Fig 3.6. Class Diagram	26
Fig 3.7. Actor	26
Fig 3.8. Lifeline	27
Fig 3.9. Reply Message	28
Fig 3.10. Found Message	28
Fig 3.11. Lost Message	29
Fig 3.12. Sequence Diagram - 1	29
Fig 3.13. Sequence Diagram - 2	30
Fig 3.14. E -R Diagram	32
Fig 4.1. System Architecture	33

## **LIST OF TABLES**

Table 1.1 Advantages of Existing System	6
Table 3.1 Face Recognition Techniques	19

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 INTRODUCTION**

Music is an important entertainment medium. With advancement of technology, the optimization of manual work has gained a lot of attention. Currently, there are many traditional music players that require songs to be manually selected and organized. Users have to create and update play-lists for each mood, which is time consuming. Some of the music players have advanced features like providing lyrics and recommending similar songs based on the singer or genre. Although some of these features are enjoyable for users, there is room to improve in the field of automation when it comes to music players. Selecting songs automatically and organizing these based on the user's mood gives user's a better experience. This can be accomplished through the system reacting to the user's emotion, saving time that would have been spent entering information manually.

Emotions can be expressed through facial expressions. For the system to understand a user's mood, we use facial expression. There are many emotion recognition systems which take captured images as input and determine the emotion. For this system, we are using a webcam and providing emotion as input for the recognition of emotion. This webcam works in the background. The system includes an algorithm that organizes songs based on the user's emotions and preferences. This algorithm suggests user's songs to play based on their emotion. The basic emotions which we have considered in our system are happy, sad, angry, neutral.

In this system there are three different modes to play songs, which take input from the user and play songs. The modes used in this system are Random mode, Queue mode, Emotion mode. These modes are selected manually by the user. Emotion mode extracts features from the user's face expression. Recognition of facial expressions is used to identify the basic human emotions. Facial expressions give important rules about emotions. A human can express his/her emotion through lip and eye. Generally, people have a large number of songs in their playlists. Thus, to avoid trouble of selecting a song, most people will just randomly select a song from their playlist and some of the songs may not be appropriate for the current mood of the user and it may disappoint the user. As a result, some of the songs are not matching the user's current emotion.

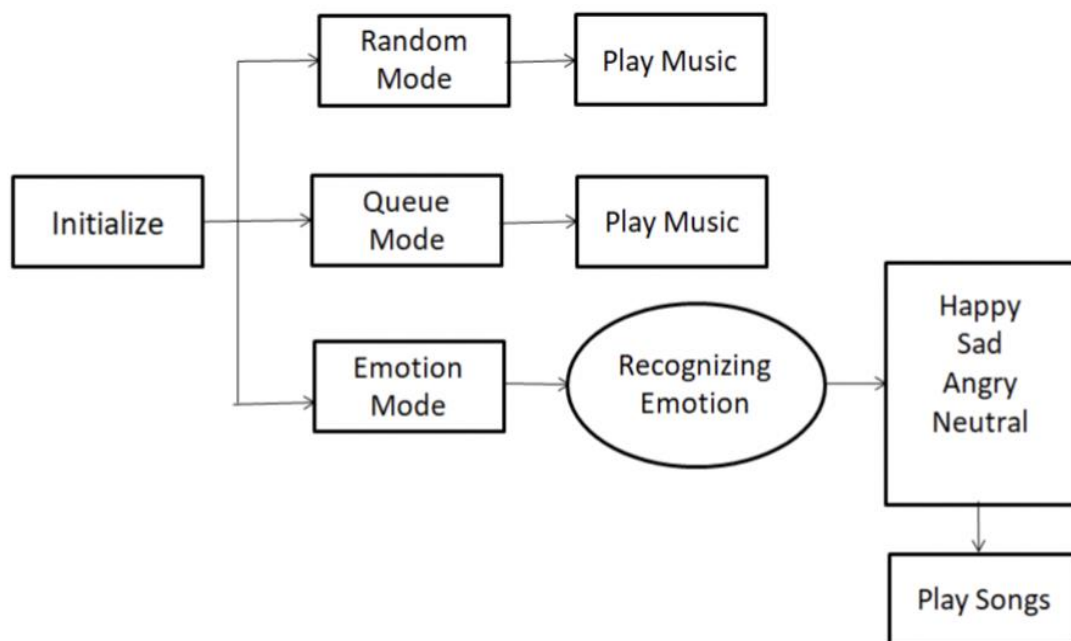
Moreover, there is no commonly used system which is able to play songs based on the current emotions of the user. Music plays a very important role in enhancing an individual's life as it is an important medium of entertainment. For music lovers and listeners and sometimes even imparts a therapeutic approach. In today's world, with ever increasing advancements in the field of multimedia and technology,

various music players have been developed with features like fast forward, reverse, and variable playback.

When we play songs based on the emotion mode the songs start playing based on user emotion and emoji is displayed based on the song i.e if user is in happy mode then happy song is played at the same time happy emoji is displayed on screen. We can also find the name of the song playing. There will be detailed information provided about the songs like Singer, Movie name and the year song released.

Although these features satisfy the user’s basic requirements, the user has to face the task of manually browsing through the playlist of songs and selecting songs based on his current mood and behaviour. Coming to Queue mode, the user should choose the mode. By selecting queue mode songs are played in order. The order of songs played is based on the playlist stored.

In the same way, the random mode is added. Here, songs are played randomly i.e when we choose random mode the songs are shuffled and played. We can play songs in the background also.



**Fig 1.1. System Overview**

## 1.2 AIM

The aim of Sentiment Analysis with Emotion Based Audio Player Using Real Time Facial Expression Extraction to provide the user an enjoyable music listening experience by serving them with sensible music content based on his/her current mood. Implementation and analysis of various viable

techniques in extracting and classifying the emotions of the user and playing music accordingly. Human face is an important organ of an individual's body and plays an important role in the extraction of an individual's emotional state. Manually segregating the list of songs and generating an appropriate playlist based on an individual's emotional feature is very tedious.

### **1.3 SCOPE**

Facial expressions are a great indicator of the state of a mind for a person. Indeed the most natural way to express emotions is through facial expressions. It is a system in which songs are played based on the mode chosen by the user. Humans tend to link the music they listen 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.

### **1.4 GOALS AND OBJECTIVES**

The main objective is that the proposed system will use face scanning and feature tracking to determine the user's mood and based on it gives a personalized playlist.

### **1.5 EXISTING SYSTEM**

The existing applications for music players are being evaluated to establish an appropriate application. There are currently many major streaming music applications with a large percentage of users. The features that are available in the previously developed Music players are as follows:

a) Manual selection of songs.

b) Party Shuffle.

c) Playlist

- Apple Music: The well-known worldwide music streaming application. The advantage of this application is the music suggestion by user favorite; however, the subscription cost is quite high.

- Spotify: Another worldwide music streaming application. It also suggests songs based on the user's data collection. Moreover, the subscription cost is much cheaper than Apple Music.

- Wynk Music: This is a music streaming & downloading app, for every mood! It has more than 1.8 million songs across Indian and International music. Stream and download songs by genres,

moods, artists or simply tune into one often many Radio channels and let it surprise

- Moodfuse - In this application, user should manually enter mood and genre that wants to be heard and mood fuse recommends the songs-list

- Steromood - User should select his mood manually by selecting the moods from the list and the application plays music from YouTube
- Musicoverly - This application has High quality songs and comprehensive music recommendations. It also suggests predefined play-list for the user



**Fig 1.2. Existing System Overview**

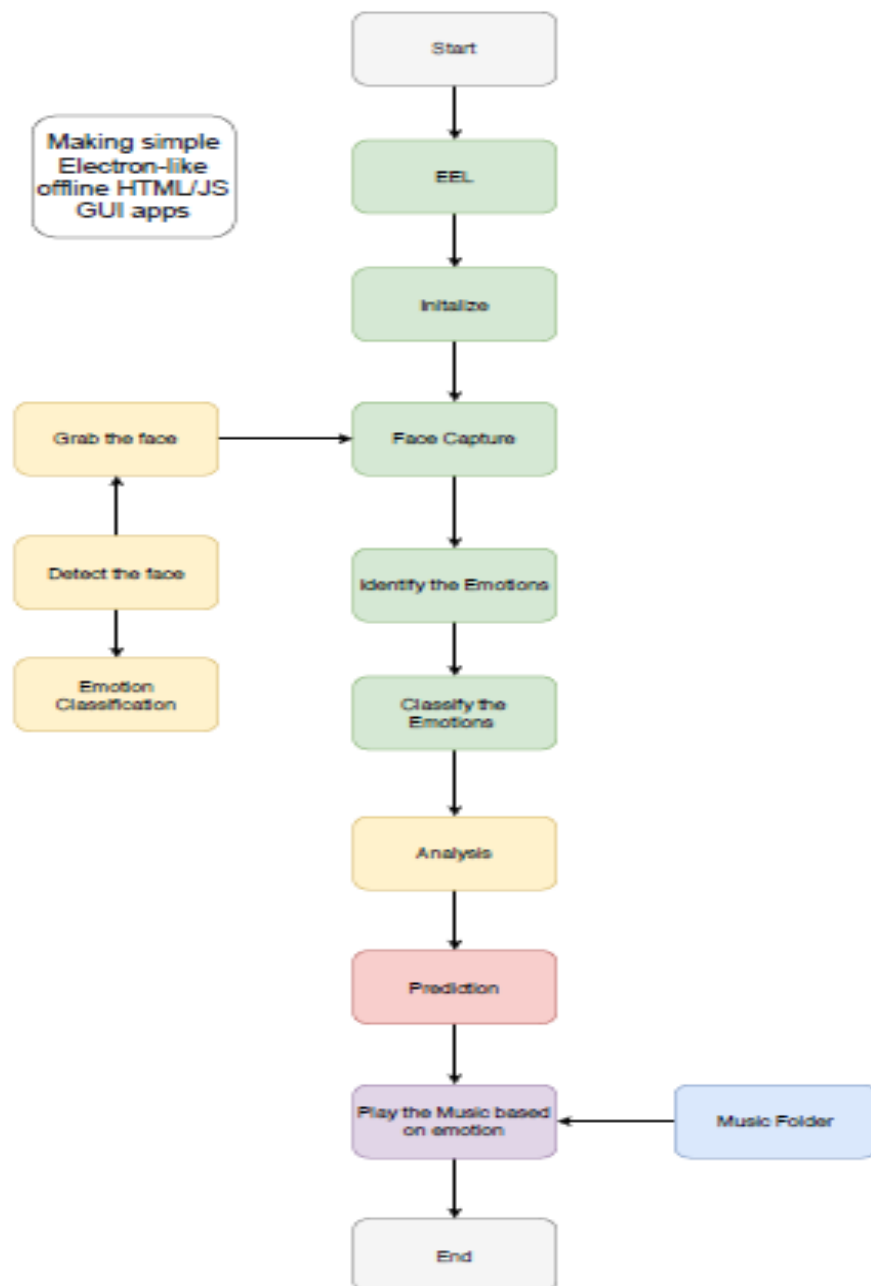
#### • **DISADVANTAGES OF EXISTING SYSTEM**

It requires the user to manually select the songs.

Randomly played songs may not match the mood of the user.

## **1.6 PROPOSED SYSTEM**

We consider the notion of collecting human emotion from the user's expressions, and explore how this information could be used to improve the user's experience with audio players. We present a new emotion based and user interactive music system. Proposed System aims to provide user preferred music based on emotion by recognising face expressions. Based on user emotions, songs related to that emotion will be played. This Proposed system has 3 types of modes for playing songs.



**Fig 1.3. Proposed System**

A user's facial expression helps the music recommendation system to decipher the current emotional state of a user and plays songs. The user needs to grant permissions to use the webcam. It is composed of mainly three modes: Random mode, Emotion mode and Queue mode. All these modes are mutually exclusive. Songs will be played in the background even if user wants to go to the other tab.

## ADVANTAGES OF EXISTING SYSTEM

	Create Playlist	Cost	Random Playing	Emotion detection
Existing System	YES	NO	YES	NO
Proposed System	YES	YES	YES	YES

**TABLE 1.1**

- Ease Of Use.
- Mobile apps can be expensive to build, maintain, and display.
- Plays Songs as per users mood.
- No trouble selecting the songs manually.
- Free of cost

## 1.7 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

### 1.7.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of funds that the company can pour into the research and development of the system is limited.



The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### **1.7.2 TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

### **1.7.3 SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

## **CHAPTER 2**

### **LITERATURE SURVEY**

Anuja Arora ; Aastha Kaul ; Vatsala Mittal [2], they submitted a program in which the DEAM data set was used to classify the emotions. It has more than 2800 songs with 4 emotions annotated: Happy, Sad, Angry and Relax, and with their values of valence and excitement. The idea behind this article is to pay attention to predicting emotions of an audio file as to how good audio elements are used in the music player.

Sushmita G. Kamble and A. H. Kulkarni [3], they proposed a system in which they used PCA(Principal component approach) for feature extraction. To classify and recognize the expression Euclidean distance classifier was used. Then, the user's corresponding emotional state is recognized. When The user's expression is recognized, songs belonging to that category are then played. They used the database with 7 expressions of 4 individual's persons that results into 112 trained images.

S. L. Happy and A. Routray [4], image from database is passed to the facial landmark detection stage to remove noise by applying Gaussian Filter or mask. Here itself they used Viola Jones technique of Haar-like features with Adaboost Learning for face detection. The feature detection stage consists of Eyebrow corners detector, Eye detector, Noise detector, Lip corner detector. After these active facial patches are extracted, The classification of features is done by SVM (Support Vector Machine). While testing it will take hundreds of images from the database and extract the features and classify them accordingly. They used CK+ (Cohn-Kanade) dataset and JAFFE dataset for training and testing the database. The training database consists of 329 images in total.

S. L. Happy and A. Routray [4], image from database is passed to the facial landmark detection stage to remove noise by applying Gaussian Filter or mask. Here itself they used Viola Jones technique of Haar-like features with Adaboost Learning for face detection. The feature detection stage consists of Eyebrow corners detector, Eye detector, Noise detector, Lip corner detector. After these active facial patches are extracted, The classification of features is done by SVM (Support Vector Machine). While testing it will take hundreds of images from the database and extract the features and classify them accordingly. They used CK+ (Cohn-Kanade) dataset and JAFFE dataset for training and testing the database. The training database consists of 329 images in total.

KrittrinChankuptarat, Raphatsak, Sriwatanaworachi, Supannada Chotipant [6],proposes a mobile music player application which is able to recommend songs based on the user emotion.When the application receives a user heart rate from a smart band or a face image from a mobile camera, it analyses what the user emotion is. Then, it suggests songs whose moods are relevant to that user's emotion. The user and song emotions in this paper are divided into four types namely: neutral, happy, sad and angry. The experimental results show that detecting the happy emotion is the most precise with around 98%, while the accuracy of the sad mood detection is the lowest with 40%.

Cyril Laurier and PerfectoHerrera, MoodCloud [7], a real-time music mood visualization tool, Mood Cloud classifies music emotions into 5 types namely: aggressive, happy, party, relax and sad. It uses the SVM library to analyze the emotion dataset. The result is then presented by using a Flash player.

## **CHAPTER 3**

### **REQUIREMENTS SPECIFICATION**

#### **3.1 DESCRIPTION OF PROBLEM**

Now -a- days people are attracted too much for entertainment. One of the ways for them to relax is by listening to songs. Due to time constraints and their busy schedule they are willing to complete all the tasks within a short span of time, but searching for the same category songs is time consuming and is a long process. The mindset of people will be like their work should be completed with less effort and the result must be within seconds.

#### **3.2 PROPOSED SOLUTION**

The current available systems require the user to do the manual selection of the songs, but the proposed system will use facial scanning and face feature tracking to determine the users mood based on it, and will provide the user with a personalized playlist, thus making the process effortless for the user. It will provide a better experience to the music connoisseurs and enthusiasts.

Along with facial scanning there are two different modes; they are random mode and queue mode based on the mode selected by the user. Using all the three modes, the above problem can be solved in a very short span of time thus decreasing the effort involved in selecting the songs manually based on the mood of the user he/she can select different types of modes.

#### **3.3 SOFTWARE REQUIREMENTS**

The software requirements document is the software specification of the system. It should include both a definition and specification of a requirements. It is a set of what the system should do rather than how it should do it. It is useful in estimating cost, planning team activities, performing tasks and tracking the team's progress throughout the development activity.

##### **3.3.1 HARDWARE REQUIREMENTS :**

Processor : Any Processor above 500 MHz.

Ram : 4 GB

Hard Disk : 4 GB

Input device : Standard Keyboard, Mouse and Integrated webcam.

Output device : High Resolution Monitor and Speakers/Headphone(High definition audio).

### **3.3.2 SOFTWARE REQUIREMENTS :**

Operating System : Windows 7 or higher.

Programming : Python 3.6 and related libraries.

### **3.3.3 SOFTWARE DESCRIPTION :**

#### **PYTHON:**

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).

It is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English words frequently whereas other languages use punctuation, and it has fewer syntactic constructions than other languages.

Some of the key advantages of learning Python:

- Python is Interpreted – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- Python is a Beginner's Language – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Python Applications: Python is a high-level general-purpose programming language that offers multiple paradigms like object-orientation, and structural and functional programming for software development.

It works on cross-platform operating systems and can be used across to develop a wide range of applications including those intended for image processing, text processing, web, and enterprise level using scientific, numeric and data from network.

#### **1. GUI-Based Desktop Applications:**

Python has simple syntax, modular architecture, rich text processing tools and the ability to work on multiple operating systems which make it a desirable choice for developing desktop based applications.

There are various GUI toolkits like wxPython, PyQt or PyGtk available which help developers create highly functional Graphical User Interface (GUI).

The various applications developed using Python includes:

- **Image Processing and Graphic Design Applications:** Python has been used to make 2D imaging software such as Inkscape, GIMP, Paint Shop Pro and Scribus. Further, 3D animation packages, like Blender, 3ds Max, Cinema 4D, Houdini, Lightwave and Maya, also use Python in variable proportions.
- **Scientific and Computational Applications:** The higher speeds, productivity and availability of tools, such as Scientific Python and Numeric Python, have resulted in Python becoming an integral part of applications involved in computation and processing of scientific data. 3D modelling software, such as FreeCAD, and finite element method software, such as Abaqus, are coded in Python.
- **Games:** Python has various modules, libraries and platforms that support development of games. For example, PySoy is a 3D game engine supporting Python 3, and PyGame provides functionality and a library for game development. There have been numerous games built using Python including Civilization-IV, Disney's Toontown Online, Vega Strike etc.

## **2. Web Frameworks and Web Applications:**

Python has been used to create a variety of web-frameworks including CherryPy, Django, TurboGears, Bottle, Flask etc. These frameworks provide standard libraries and modules which simplify tasks related to content management, interaction with databases and interfacing with different internet protocols such as HTTP, SMTP, XML-RPC, FTP and POP. Plone, a content management system; ERP5, an open source ERP which is used in aerospace, apparel and banking; Odoo – a consolidated suite of business applications; and Google App engine are a few of the popular web applications based on Python

## **3. Enterprise and Business Applications:**

With features that include special libraries, extensibility, scalability and easily readable syntax, Python is a suitable coding language for customizing larger applications. Reddit, which was originally written in Common Lisp, was rewritten in Python in 2005. Python also contributed in a large part to functionality in YouTube.

## **4. Operating Systems:**

Python is often an integral part of Linux distributions. For instance, Ubuntu's Ubiquity Installer, and Fedora's and Red Hat Enterprise Linux's Anaconda Installer are written in Python. Gentoo Linux makes use of Python for Portage, its package management system.

Features of Python There are many features in Python, some of which are discussed below –

### **1.Easy to code:**

Python is a high -level programming language. Python is very easy to learn language as compared to other language like c, c#, java script, java etc. It is very easy to code in python language and anybody can learn python basic in few hours or days. It is also a developer friendly language.

### **2. Free and Open Source:**

Python language is freely available at the official website and we can download it easily.

### **3.Object-Oriented Language:**

One of the key features of python is Object-Oriented programming. Python supports object oriented language and concepts of classes, object encapsulation etc.

**4. GUI Programming Support:** Graphical Users interfaces can be made using a module such as PyQt5, PyQt4, wxPython or Tk in python. PyQt5 is the most popular option for creating graphical apps with Python.

**5. High-Level Language:** Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

**6.Extensible feature:** Python is an Extensible language. We can write some python code into c or c++ language and also, we can compile that code in c/c++ language.

**7. Python is Portable language:** Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on another platform such as Linux, Unix and Mac then we do not need to change it, we can run this code on any platform

**8. Python is an Integrated language:** Python is also an Integrated language because we can easily integrate python with other languages like c, c++ etc.

**9. Interpreted Language:** Python is an Interpreted Language. because python code is executed line by line at a time. Unlike other languages c, c++, java etc there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called bytecode.

### **10. Dynamically Large Standard:**

Library Python has a large standard library which provides a rich set of modules and functions so you do not have to write your own code for every single thing. There are many libraries present in python such as regular expressions, unit-testing, web browsers etc. 11.Dynamically Typed Language: Python is a dynamically-typed language. That means the type (for example- int, double, long etc) for a variable is decided at run time not in advance. Because of this feature we don't need to specify the type of variable.

## **OpenCV:**

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. OpenCV is the huge open-source library for computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing.

Applications of OpenCV: There are lots of applications which are solved using OpenCV, some of them are listed below.

- face recognition
- Automated inspection and surveillance
- number of people – count (foot traffic in a mall, etc)
- Vehicle counting on highways along with their speeds
- Interactive art installations
- Anomaly (defect) detection in the manufacturing process (the odd defective products)
- Street view image stitching
- Video/image search and retrieval
- Robot and driver-less car navigation and control
- object recognition
- Medical image analysis
- Movies – 3D structure from motion
- TV Channels advertisement recognition



**Time:**

The Python time module represents time-based objects in Python. Developers use the time() function to return the current time as a UNIX timestamp. The ctime() function translates a UNIX timestamp into a standard year, month, day, and time format.

**OS:**

The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system dependent functionality. The `*os*` and `*os.path*` modules include many functions to interact with the file system.

**GLOB:**

In Python, the glob module is used to retrieve files/path names matching a specified pattern. The pattern rules of glob follow standard Unix path expansion rules. It is also predicted that according to benchmarks it is faster than other methods to match path names in directories.

**ARGPARSE:**

The Python argparse library was released as part of the standard library with Python 3.2 on February the 20th, 2011. It was introduced with Python Enhancement Proposal 389 and is now the standard way to create a CLI in Python, both in 2.7 and 3.2+ versions.

The Python argparse library:

- Allows the use of positional arguments
- Allows the customization of the prefix chars
- Supports variable numbers of parameters for a single option
- Supports subcommands (A main command line parser can use other command line parsers depending on some arguments.)

**ELL:**

- Eel is a little Python library for making simple.Electron-like offline HTML/JS GUI apps, with full access to Python capabilities and libraries.
- Eel hosts a local webserver, then lets you annotate functions in Python so that they can be called from Javascript, and vice versa.

- Eel is designed to take the hassle out of writing short and simple GUI applications.

## **WINSOUND:**

The winsound module provides access to the basic sound-playing machinery provided by Windows platforms. It includes functions and several constants:-

- `winsound.Beep(frequency, duration):`

Beep the PC's speaker. The frequency parameter specifies frequency, in hertz, of the sound, and must be in the range 37 through 32,767. The duration parameter specifies the number of milliseconds the sound should last. If the system is not able to beep the speaker, `RuntimeError` is raised.

- `winsound.PlaySound(sound, flags):`

Call the underlying `PlaySound()` function from the Platform API. The sound parameter may be a filename, audio data as a string, or `None`. Its interpretation depends on the value of flags, which can be a bitwise OR ed combination of the constants described below. If the sound parameter is `None`, any currently playing waveform sound is stopped. If the system indicates an error, `RuntimeError` is raised.

- `winsound.MessageBeep(type=MB_OK):`

Call the underlying `MessageBeep()` function from the Platform API. This plays a sound as specified in the registry. The type argument specifies which sound to play; possible values are `-1`, `MB_ICONASTERISK`, `MB_ICONEXCLAMATION`, `MB_ICONHAND`, `MB_ICONQUESTION`,

and `MB_OK`, all described below. The value `-1` produces a "simple beep"; this is the final fallback if a sound cannot be played otherwise.

- `winsound.SND_ALIAS:`

The sound parameter is a sound association name from the registry. If the registry contains no such name, play the system default sound unless `SND_NODEFAULT` is also specified. If no default sound is registered, raise `RuntimeError`.

### **3.3.4 FUNCTIONAL REQUIREMENTS :**

Functional requirements are statement of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situation.:

- Learn and identify image capture by webcam.
- Extracting and analyzing the emotion.

### **3.3.5 NON FUNCTIONAL REQUIREMENTS :**

The following is a list of non-functional requirements. The specific details will need to be defined by internal stakeholders.

#### **3.3.5.1 Functionality :**

The application is developed in such a way that any future enhancement can be easily implementable. The project is developed in such a way that it requires minimal maintenance. The software used is open source and easy to install. The application developed should be easy to install and use.

#### **3.3.5.2 Reliability:**

It is the maturity, fault tolerance and recoverability. The system is reliable for any number of user input and training dataset. Though we have limited the training dataset for one class to be 300.

#### **3.3.5.3 Usability:**

It is easy to understand, learn and operate the software system. The user can upload data and predict the suitable crop.

#### **3.3.5.4 Safety :**

Safety-critical is an issue associated with its integrity level. The computer system being used is protected by a password.

#### **3.3.5.5 Security :**

It does not block the some available ports through the Windows firewall. The web camera port should be enabled automatically, otherwise the user must enable it every time.

#### **3.3.5.6 Robustness:**

The application is developed in such a way that any future enhancement can be easily implementable. The project is developed in such a way that it requires minimal maintenance. The software used is open source and easy to install. The application developed should be easy to install and use.

### **3.3.5.7 Communications:**

The application is developed in such a way that communication can be handled through sign language for deaf people. Similarly sign language to voice for dumb people. 3.3.4.9 Software Analysis The project primarily focuses on sign language detection. We implemented it with the Python 3.6 version. The libraries required are to be installed prior to executing the project. We installed CV2 for OpenCV.

### **3.3.6 MODULES :**

The proposed system can be achieved by implementing the following modules

- Image Capturing
- Emotion Extraction
- Audio Integration
- Generating Playlist

#### **3.3.6.1 Image Capturing:**

The audio player appears and opens the device's camera to capture the emotion. All these processes are carried out in the backside of the music player.



#### **3.3.6.2 Emotion Extraction:**

A picture is taken throughout the runtime of the appliance, which once preprocessing, is foreseen to belong to one of the emotion categories by the fisher face classifier. The model additionally permits the user to customize the model so as to cut back the variance among the categories any, initio or sporadically, such that solely variance would be that of feeling modification.

### **Fisherfaces method:**

Discriminant Analysis is the best approach to scale back dimensionality especially in classification domain. Linear discriminant analysis is one among the foremost popular Discriminant Analysis. It's a way to find a linear combination of features which may be wont to separate classes of various objects. It can thus be used for dimensionality reduction as well as a linear classifier.

Fisherfaces method using fisher's Linear Discriminant (FLD) provides higher between-class scatter as compared to PCA. It focuses on maximizing the ratio of between-class to within-class scatter. As a result of which the tightly spaced clusters are formed.

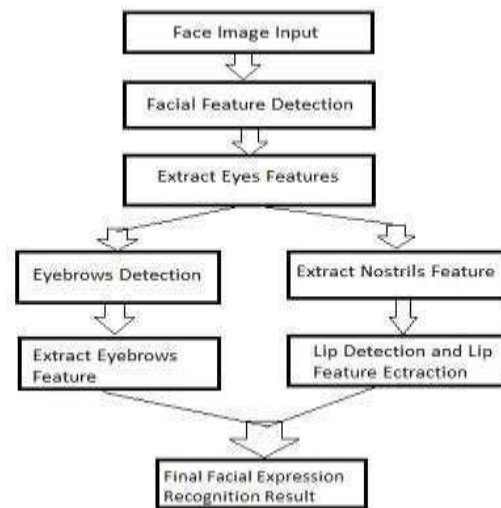
As the data set that's used for learning is labeled, this information is used to create a more reliable model to scale back dimensionality. It starts by creating a picture matrix wherein each column may be a vector that represents pixel intensities of the image.

There are many factors that contribute to conveying the emotions of a patient. Pose, speech, facial expressions, behavior and actions are a number of them. From these above mentioned factors, facial expressions have a better importance since they're easily perceptible.

<b>Parts of the Face</b>	<b>Emotion</b>
lip corner pulled, open eyes, open mouth, cheeks raised	Happy
Eyebrow pulled down, open eyes, lip tightened	Anger
Outer eyebrow down, inner eyebrows raised, eyes closed, lip corner down	Sad
Lip corner depressor, lower lip depressor, eyebrows down, nose wrinkled	Neutral

**TABLE 3.1**

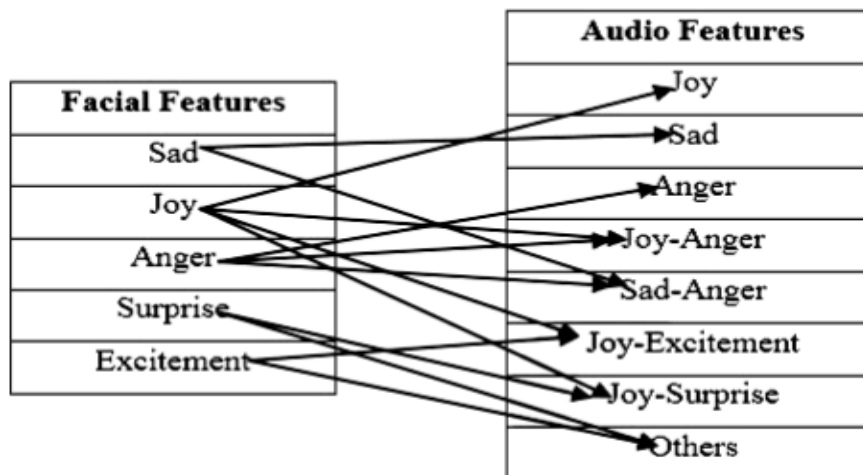
Above mentioned table is giving a brief idea of different facial parts relating to different types of Emotion/Expression carried by Humans.



**Fig 3.1. Face feature extraction**

### 3.3.6.3 Audio Integration:

Emotions extracted for the songs are stored as a meta-data in the database. Mapping is performed by querying the meta-data database. The emotion extraction module and audio feature extraction module is finally mapped and combined using an Emotion-Audio integration module. For example, if an input facial image is categorized under joy, the system will display songs under joy, joy-anger, Joy-Excitement, Joy-surprise category.



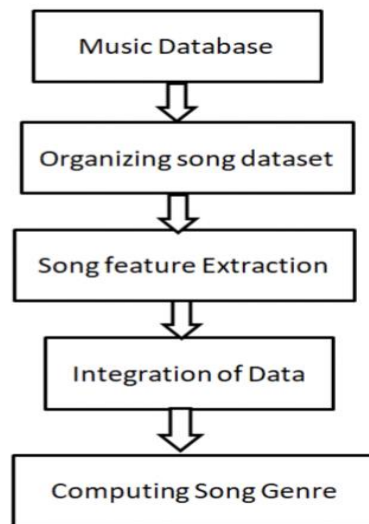
**Fig 3.2. Integration of emotion**

### 3.3.6.4 Generating playlist:

Categorizing music files consistent with their genre may be a challenging task within the area of music information retrieval (MIR). With the expansion of online music databases and easy access to music content, therapist find it increasingly hard to manage the songs that improves patients health, a method to categorize

and organize songs is based on the genre, which is identified by some characteristics of the music like rhythmic structure, harmonic content and instrumentation.

A number of emotions is collected and put up in the list. Each emotion category has a number of songs listed in it.



**Fig 3.3. Generating playlist**

### 3.4 UML DIAGRAMS

Basically UML stands for Unified Modeling Language. UML was created by the Object Management Group (OMG). It is a standardized modeling language used by software developers and quite different from programming languages like Python, C and C++.UML consists of an integrated set of diagrams and provides users with ready-to-use. These diagrams are developed to help system and software developers for specifying, visualizing, constructing and documenting the artifacts of software systems, non - software systems and for business modeling as well.

UML diagrams are divided into 3 different categories. Such as,

- Structural Diagram
- Behavioral Diagram
- Integration Diagram

Diagrams proposed in this project are class, use case, activity and sequence diagrams. These are divided into 2 methods.

## ❖ **SYSTEM ANALYSIS METHODS**

- **USE CASE DIAGRAM**
- **ACTIVITY DIAGRAM**

## ❖ **SYSTEM DESIGN METHODS**

- **CLASS DIAGRAM**
- **SEQUENCE DIAGRAM**

### **3.5 SYSTEM ANALYSIS METHODS**

#### **3.5.1 USE CASE DIAGRAM:**

To model a system, the most important aspect is to capture the dynamic behaviour. Dynamic behavior means the behaviour of the system when it is running/operating. Only static behavior is not sufficient to model a system; rather, dynamic behavior is more important than static behavior. There should be some internal or external factors for making the interaction. These internal and external agents are known as actors. Use case diagrams consist of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. Use case diagrams are employed in UML (Unified Modelling Language), a standard notation for the modelling of real-world objects and systems. System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer service- oriented task..

A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system, are defined according to their roles.
- The use cases, which are the specific roles played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

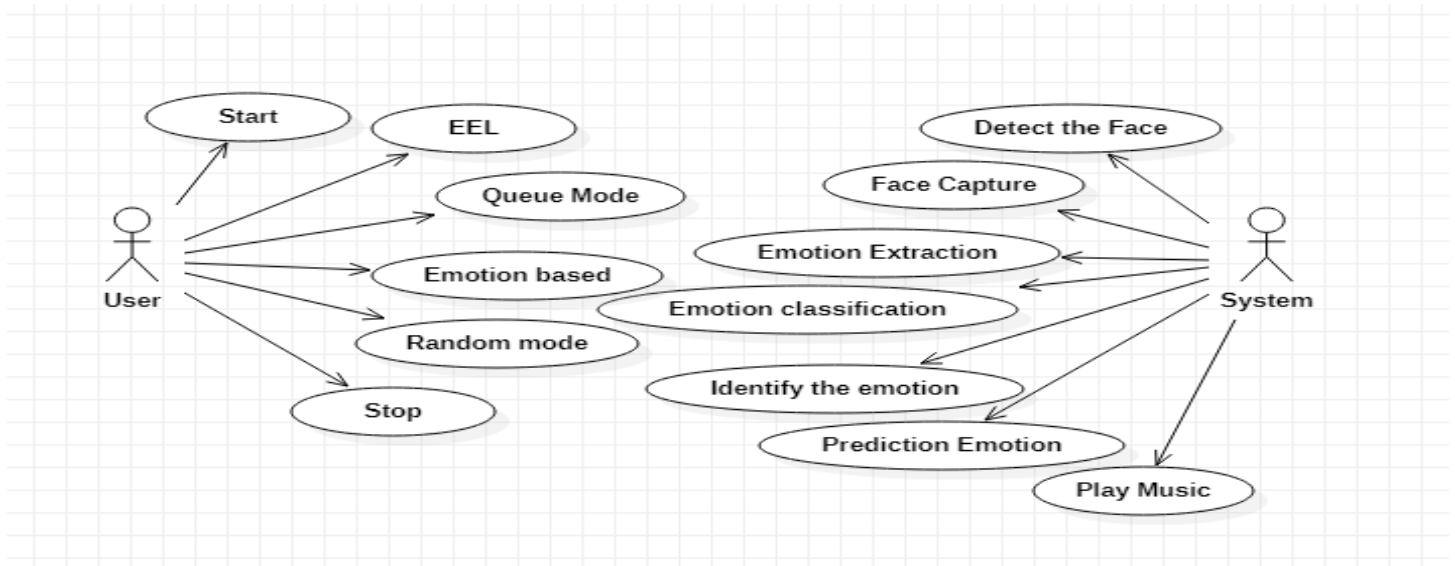
A Use Case model can be developed by following the steps below.

1. Identify the Actors (role of users) of the system.
2. For each category of users, identify all roles played by the users relevant to the system.



3. Identify what are the users required for the system to be performed to achieve these goals.
4. Create use cases for every goal.
5. Structure the use cases.
6. Prioritize, review, estimate and validate the users.

The actors in the proposed system are user and system. Users can get access to the system and can use all the three features of the system. System is another actor who is responsible for extracting, analyzing and classifying the emotion of the user.



**Fig 3.4. USE CASE Diagram**

### 3.5.2 ACTIVITY DIAGRAM:

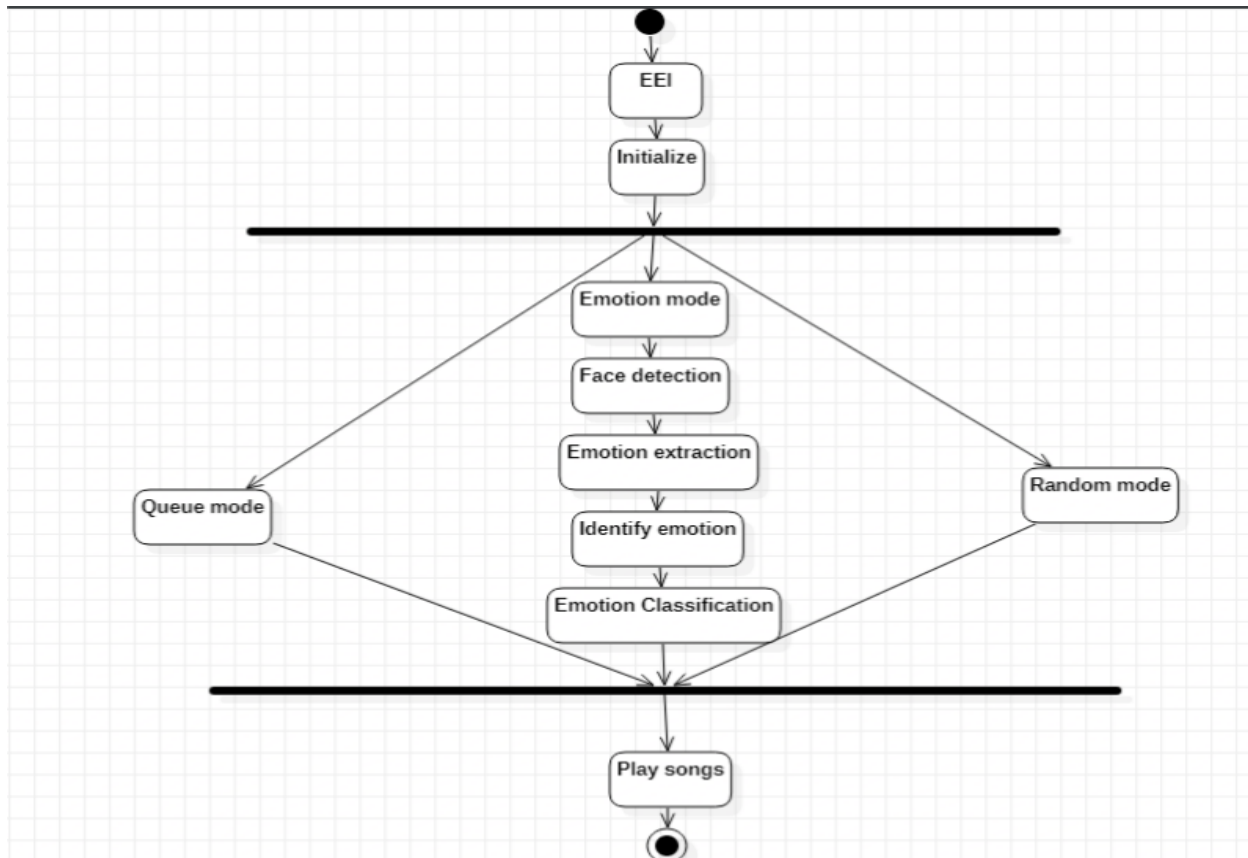
Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent.

The basic purpose of activity diagrams is similar to the other four diagrams. It captures the dynamic behaviour of the system. Other four diagrams are used to show the message flow from one object to another but the activity diagram is used to show the message flow from one activity to another. Activity is a particular operation of the system.

The purpose of an activity diagram can be described as –

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

Initially we start the system and then select the mode of playing. There are 3 available modes in this system. By choosing the mode, songs start playing.



**Fig 3.5. Activity Diagram**

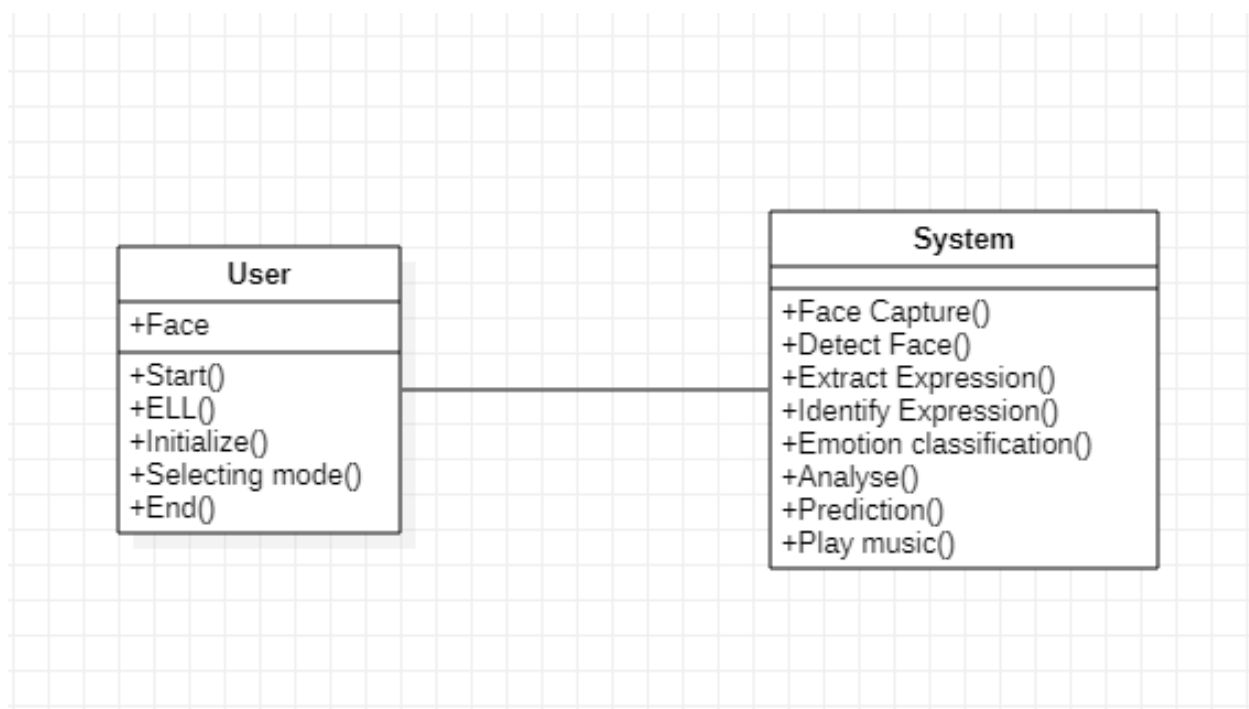
## **3.6 SYSTEM DESIGN METHODS**

### **3.6.1 CLASS DIAGRAM:**

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Class diagram shows a collection of classes, interfaces, associations,

collaborations, and constraints. It is also known as a structural diagram. The purpose of the class diagram can be

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.



**Fig 3.6. Class Diagram**

### 3.6.2 SEQUENCE DIAGRAM:

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function? These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations –

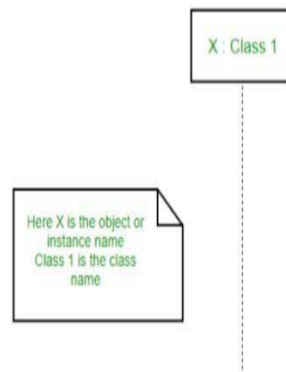
**Actors** – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram.



**Fig 3.7. Actor**

We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.

**Lifelines** – A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically, each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram. The standard in UML for naming a lifeline follows the following format –  
Instance Name : Class Name



**Fig 3.8. Lifeline**

We display a lifeline in a rectangle called head with its name and type. The head is located on top of a vertical dashed line (referred to as the stem) as shown above. If we want to model an unnamed instance, we follow the same pattern except now the portion of lifeline's name is left blank.

**Difference between a lifeline and an actor** – A lifeline always portrays an object internal to the system whereas actors are used to depict objects external to the system. The following is an example of a sequence diagram.

**Messages** – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages from the core of a sequence diagram. Messages can be broadly classified into the following categories:

A sequence diagram with different types of messages

**Synchronous messages** – A synchronous message waits for a reply before the interaction can move forward. The sender waits until the receiver has completed the processing of the message. The caller

continues only when it knows that the receiver has processed the previous message

i.e. it receives a reply message. A large number of calls in object-oriented programming is synchronous. We use a solid arrow head to represent a synchronous message.

**Asynchronous Messages** – An asynchronous message does not wait for a reply from the receiver. The interaction moves forward irrespective of the receiver processing the previous message or not. We use a lined arrow head to represent an asynchronous message.

**Create message** – We use a Create message to instantiate a new object in the sequence diagram. There are situations when a particular message call requires the creation of an object. It is represented with a dotted arrow and a word labelled on it to specify that it is the create Message symbol.

**Delete Message** – We use a Delete Message to delete an object. When an object is deallocated memory or is destroyed within the system, we use the Delete Message symbol. It destroys the occurrence of the object in the system. It is represented by an arrow terminating with a x.

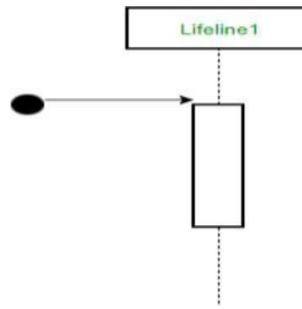
**Self-Message** – Certain scenarios might arise where the object needs to send a message to itself. Such messages are called Self Messages and are represented with a U-shaped arrow.

**Reply Message** – Reply messages are used to show the message being sent from the receiver to the sender. We represent a return/reply message using an open arrowhead with a dotted line. The interaction moves forward only when a reply message is sent by the receiver.



**Fig 3.9: Reply message**

**Found Message** – A Found message is used to represent a scenario where an unknown source sends the message. It is represented using an arrow directed towards a lifeline from an endpoint. For example: Consider the scenario of a hardware failure.

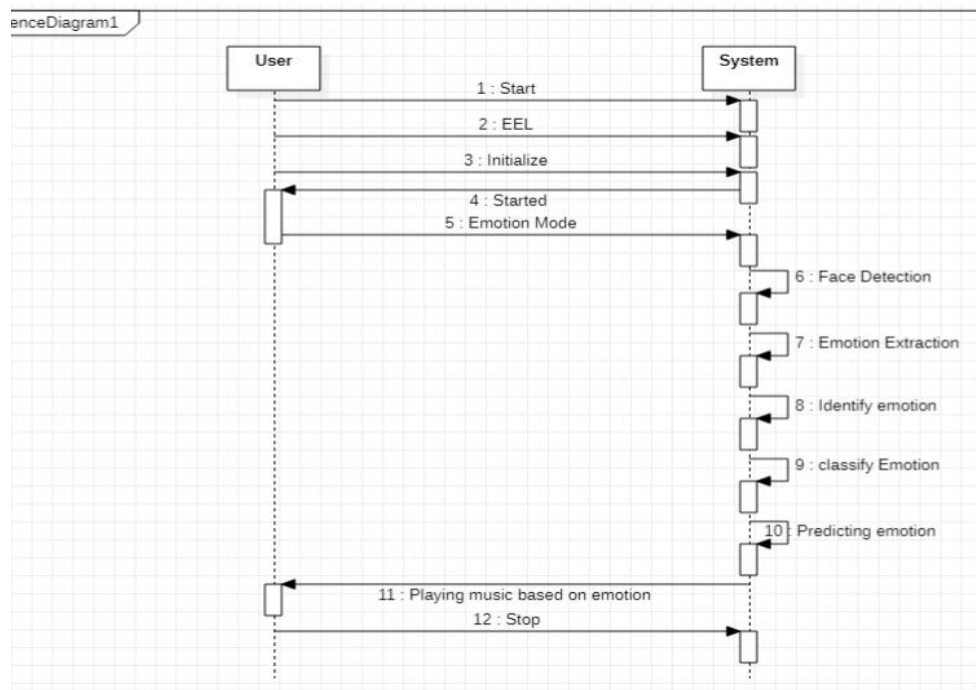


**Fig 3.10: Found message**

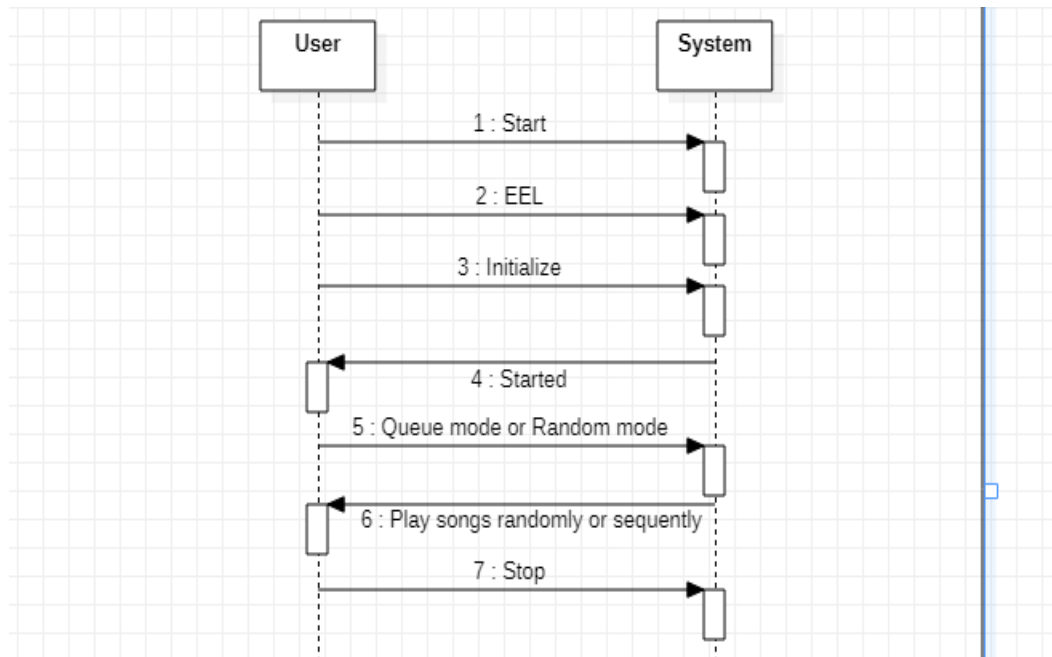
**Lost Message** – A Lost message is used to represent a scenario where the recipient is not known to the system. It is represented using an arrow directed towards an end point from a lifeline. For example: Consider a scenario where a warning is generated.



**Fig 3.11: Lost message**



**Fig 3.12: Sequence Diagram-1**



**Fig 3.13: Sequence Diagram-2**

### 3.7 E-R DIAGRAM

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of the E-R model are: entity set and relationship set.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in a database, so by showing relationships among tables and their attributes, ER diagrams show the complete logical structure of a database. Let’s have a look at a simple ER diagram to understand this concept.

Here are the geometric shapes and their meaning in an E-R Diagram.

**Rectangle:** Represents Entity sets.

**Ellipses:** Attributes

**Diamonds:** Relationship Set

**Lines:** They link attributes to Entity Sets and Entity sets to Relationship Set

**Double Ellipses:** Multivalued Attributes **Dashed Ellipses:** Derived Attributes **Double Rectangles:** Weak Entity Sets

**Double Lines:** Total participation of an entity in a relationship set. An ER diagram has three main components:

1. Entity
2. Attribute
3. Relationship.

### 1. Entity:

An entity is an object or component of data. An entity is represented as a rectangle in an ER diagram.

#### **Weak Entity:**

An entity that cannot be uniquely identified by its own attributes and relies on the relationship with another entity is called a weak entity. The weak entity is represented by a double rectangle.

### 2. Attribute:

An attribute describes the property of an entity. An attribute is represented as Oval in an ER diagram. There are four types of attributes:

1. Key attribute
2. Composite attribute
3. Multi valued attribute
4. Derived attribute

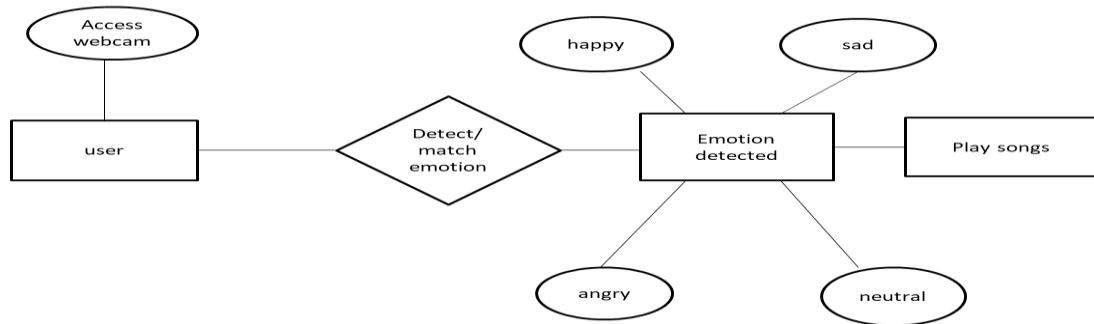
### 3. Relationship:

A relationship is represented by a diamond shape in the ER diagram, it shows the relationship among entities. There are four types of relationships:

1. One to one
2. One to many



3. Many to one
4. Many to many

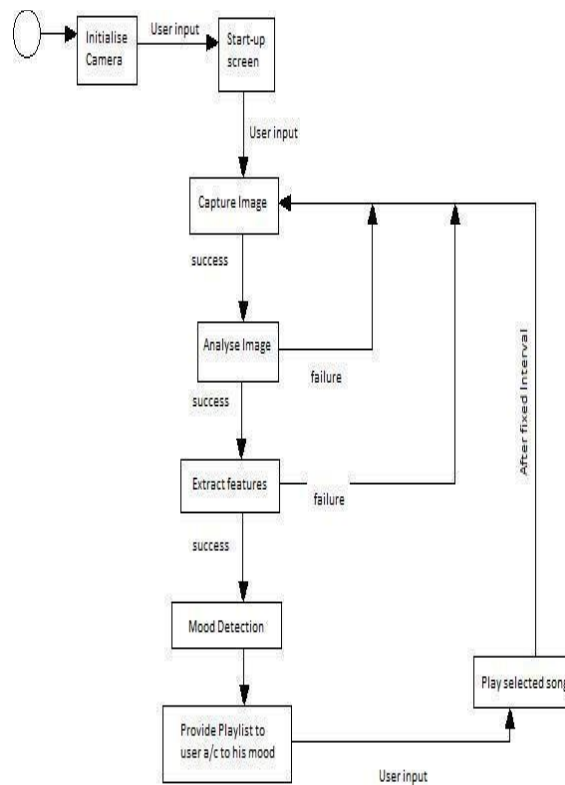


**Fig 3.12: E-R Diagram**

## CHAPTER 4

### SYSTEM DESIGN

#### 4.1 SYSTEM ARCHITECTURE



**Fig 4.1: System Architecture**

Image processing and use of facial detection are the two main cornerstones of the proposed system. The input model for the system are still images of the user, which are used for further processing for determining the mood of the user. The proposed system will capture an image of the respective user at the initialization of the application.

System for the emotion recognition is divided into 3 parts: face location determination, emotion classification and feature extraction. The project and system heavily focuses on the latter two parts, the feature extraction and classification. The extracted data is fed to a classifier which determines the emotion of the user and plays the song.

## CHAPTER 5

### SYSTEM IMPLEMENTATION

#### 5.1 TOOLS & TECHNIQUES

##### Visual Studio

Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging. Visual Studio Code features a lightning fast source code editor, perfect for day-to-day use. With support for hundreds of languages, VS Code helps you be instantly productive with syntax highlighting, bracket matching, auto-indentation, box-selection, snippets, and more. Intuitive keyboard shortcuts, easy customization and community-contributed keyboard shortcut mappings let you navigate your code with ease. For serious coding, you'll often benefit from tools with more code understanding than just blocks of text.

Visual Studio Code includes built-in support for IntelliSense code completion, rich semantic code understanding and navigation, and code refactoring. And when the coding gets tough, the tough get debugging. Debugging is often the one feature that developers miss most in a leaner coding experience, so we made it happen. Visual Studio Code includes an interactive debugger, so you can step through source code, inspect variables, view call stacks, and execute commands in the console. VS Code also integrates with build and scripting tools to perform common tasks making everyday workflows faster. VS Code has support for Git so you can work with source control without leaving the editor including viewing pending changes diffs.

##### PYTHON

##### INSTALLATION STEPS OF PYTHON:

To install Python, you need to download the official Python executable installer. Next, you need to run this installer and complete the installation steps. Finally, you can configure the PATH variable to use python from the command line.

##### Step 1: Download the Python Installer binaries

- Open the official Python website in your web browser. Navigate to the Downloads tab for Windows.

- Choose the latest Python 3 release. In our example, we choose the latest Python 3.7.3 version. Click on the link to download Windows x86 executable installer if you are using a 32-bit installer.
- In case your Windows installation is a 64-bit system, then download Windows x86-64 executable installer.

## Step 2: Run the Executable Installer

- Once the installer is downloaded, run the Python installer.
- Check the Install launcher for all users. Further, you may check the Add Python 3.7 to path check box to include the interpreter in the execution path.
- Select **Customize installation**.



Choose the optional features by checking the following check boxes:

1. Documentation
2. pip
3. tcl/tk and IDLE (to install tkinter and IDLE)
4. Python test suite (to install the standard library test suite of Python)
5. Install the global launcher for `.py` files. This makes it easier to start Python
6. Install for all users. Click Next.

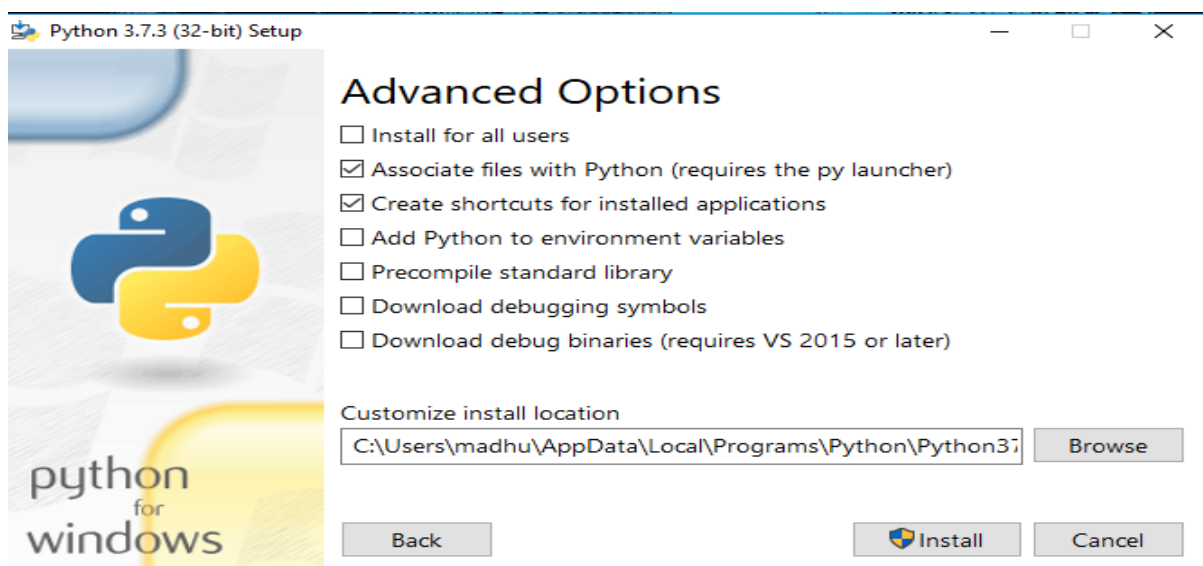
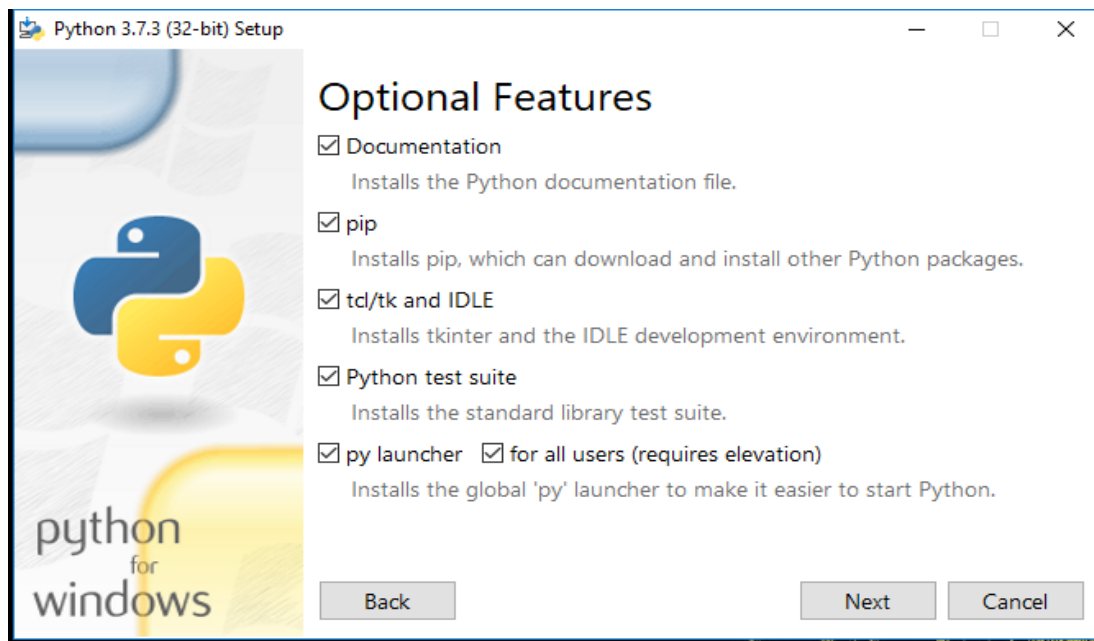
This takes you to Advanced Options available while installing Python. Here, select the Install for all users and Add Python to environment variables check boxes.

Optionally, you can select the Associate files with Python, Create shortcuts for installed applications and other advanced options. Make note of the python installation directory displayed in this step. You would

need it for the next step.

After selecting the Advanced options, click Install to start installation.

1. Once the installation is over, you will see a Python Setup Successful window.



### Step 3: Add Python to environmental variables

The last (optional) step in the installation process is to add Python Path to the System Environment variables. This step is done to access Python through the command line. In case you have added Python to environment variables while setting the Advanced options during the installation procedure, you can avoid this step. Else, this step is done manually as follows.

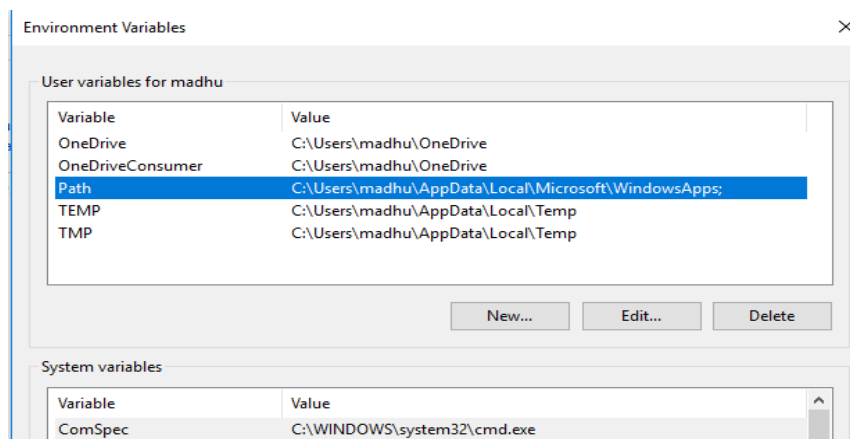
In the Start menu, search for “advanced system settings”. Select “View advanced system settings”. In the

“System Properties” window, click on the “Advanced” tab and then click on the “Environment Variables” button.

Locate the Python installation directory on your system. If you followed the steps exactly as above, python will be installed in below locations:

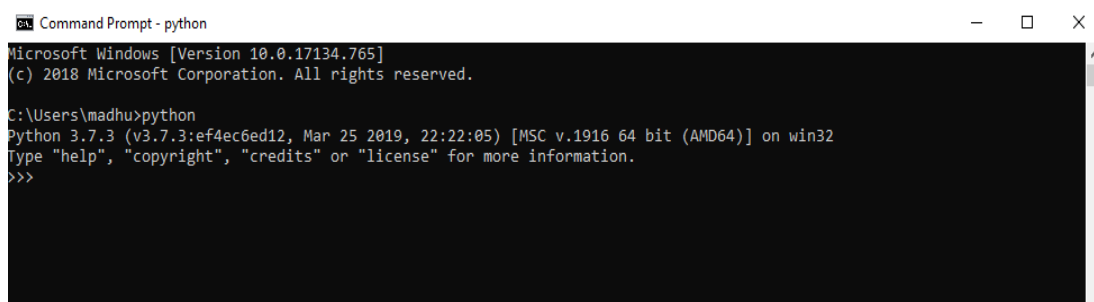
- C:\Program Files (x86)\Python 37-32: for 32-bit installation
- C:\Program Files\Python 37-32: for 64-bit installation

The folder name may be different from “Python37-32” if you installed a different version. Look for a folder whose name starts with Python.

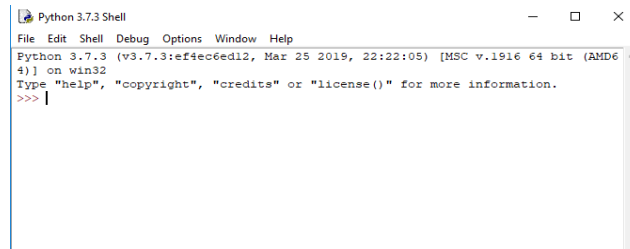


#### Step 4: Verify the Python Installation

You have now successfully installed Python 3.7.3 on Windows 10. You can verify if the Python installation is successful either through the command line or through the IDLE app that gets installed along with the installation. Search for the command prompt and type “python”. You can see that Python 3.7.3 is successfully installed.



An alternate way to reach python is to search for “Python” in the start menu and click on IDLE (Python 3.7 64-bit). You can start coding in Python using the Integrated Development Environment(IDLE).



## APPLICATIONS OF PYTHON

As mentioned before, Python is one of the most widely used languages over the web. I'm going to list few of them here:

**Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

**Easy-to-read** – Python code is more clearly defined and visible to the eyes.

**Easy-to-maintain** – Python's source code is fairly easy-to-maintain.

**A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

**Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

**Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

**Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

**Databases** – Python provides interfaces to all major commercial databases.

**GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

**Scalable** – Python provides a better structure and support for large programs than shell scripting.

### **Facial emotion detection:**

Input for this system is captured using a webcam and after taking input image is pre-processed, where image is converted into grayscale. Emotion is extracted using a training model where the image undergoes various techniques which focus on certain features. In this system, the FisherFace method is used to extract the

features. OpenCV is used to detect the face and features in the face. Once the emotion is extracted, the process will wait until the same emotion continues for a certain number of times and probabilities are also considered. After the emotion is recognized, the user can listen to the songs.

## 5.2 IMPLEMENTATION CODE

### Capture.py

```
import cv2
import argparse
import time
import os
import Update_Model
import glob
import random
import eel
import light
#import winsound

frequency=2500
duration=1000

eel.init('WD')
emotions=["angry", "happy", "sad", "neutral"]
fishface = cv2.face.FisherFaceRecognizer_create()
font = cv2.FONT_HERSHEY_SIMPLEX
"""try:
fishface.load("model.xml")
except:
print("No trained model found... --update will create one.")"""

parser=argparse.ArgumentParser(description="Options for emotions based music player(Updating the model)")
parser.add_argument("--update", help="Call for taking new images and retraining the model.", action="store_true")
args=parser.parse_args()
facedict={}
video_capture=cv2.VideoCapture(0)
facecascade=cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
def crop(clahe_image, face):
for (x, y, w, h) in face:
faceslice=clahe_image[y:y+h, x:x+w]
faceslice=cv2.resize(faceslice, (350, 350))
facedict["face%s" %(len(facedict)+1)]=faceslice
return faceslice
```



```

def grab_face():
#ret, frame=video_capture.read()
ret, frame=light.nolight()
#cv2.imshow("Video", frame)
cv2.imwrite('test.jpg', frame)
cv2.imwrite("images/main%s.jpg" %count, frame)
gray=cv2.imread('test.jpg',0)
#gray=cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
clahe=cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
clahe_image=clahe.apply(gray)
return clahe_image

def detect_face():
clahe_image=grab_face()
face=facecascade.detectMultiScale(clahe_image,scaleFactor=1.1,minNeighbors=15,minSize=(10,10),
flags=cv2.CASCADE_SCALE_IMAGE)
if len(face)>=1:
faceslice=crop(clahe_image, face)
#return faceslice
else:
print("No/Multiple faces detected!!, passing over the frame")

def save_face(emotion):
print("\n\nLook "+emotion+" until the timer expires and keep the same emotion for some time.")
#winsound.Beep(frequency, duration)
print('\a')

for i in range(0, 5):
print(5-i)
time.sleep(1)

while len(facedict.keys())<16:
detect_face()

for i in facedict.keys():
path, dirs, files = next(os.walk("dataset/%s" %emotion))
file_count = len(files)+1
cv2.imwrite("dataset/%s/%s.jpg" %(emotion, (file_count)), facedict[i])
facedict.clear()

def update_model(emotions):
print("Update mode for model is ready")
checkForFolders(emotions)

```

```

for i in range(0, len(emotions)):
    save_face(emotions[i])
    print("Collected the images, looking nice! Now updating the model...")
    Update_Model.update(emotions)
    print("Model train successful!!")

```

```

def checkForFolders(emotions):
    for emotion in emotions:
        if os.path.exists("dataset/%s" %emotion):
            pass
        else:
            os.makedirs("dataset/%s" %emotion)

```

```

def identify_emotions():
    prediction=[]
    confidence=[]

```

```

for i in facedict.keys():
    pred, conf=fishface.predict(facedict[i])
    cv2.imwrite("images/%s.jpg" %i, facedict[i])
    prediction.append(pred)
    confidence.append(conf)
    output=emotions[max(set(prediction), key=prediction.count)]
    print("You seem to be %s" %output)
    facedict.clear()
    return output;
    #songlist=[]
    #songlist=sorted(glob.glob("songs/%s/*" %output))
    #random.shuffle(songlist)
    #os.startfile(songlist[0])
    count=0
    @eel.expose
    def getEmotion():

```

```

    count=0
    while True:
        count=count+1
        detect_face()
        if args.update:
            update_model(emotions)
            break
        elif count==10:
            fishface.read("model2.xml")

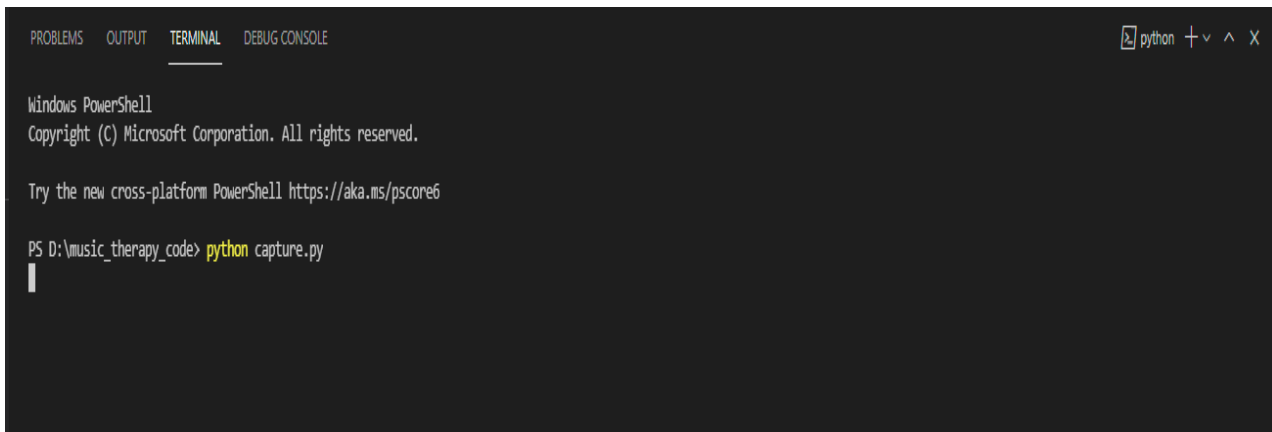
```

```
return identify_emotions()
break

eel.start('main.html')
```

## 5.3 SCREENSHOTS

The terminal window is opened to run the program.



The screenshot shows a terminal window within Visual Studio Code. The terminal title bar indicates it is a 'python' session. The output shows the Windows PowerShell prompt, copyright information for Microsoft Corporation, and a suggestion to try the cross-platform PowerShell. The command prompt shows the user is in the directory 'D:\music\_therapy\_code' and has executed the command 'python capture.py'.

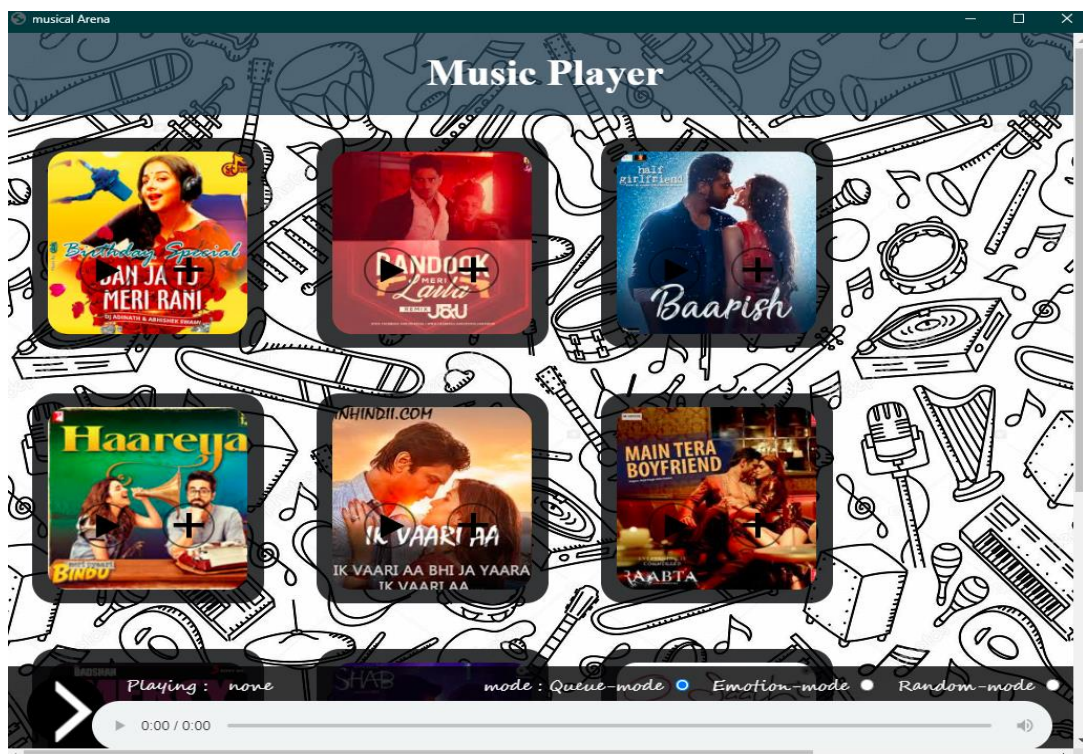
```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE python + ^ X

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\music_therapy_code> python capture.py
```

After Executing the above command(python capture.py) in the terminal window in Visual studio code Tool, an audio player opens in a new window. By default it enables queue mode.



## **CHAPTER 6**

### **SYSTEM TESTING**

#### **6.1 PURPOSE OF TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement. The major objectives of Software testing are as follows:

- Finding defects which may get created by the programmer while developing the software.
- Gaining confidence in and providing information about the level of quality.
- To prevent defects.
- To make sure that the end result meets the business and user requirements.
- To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
- To gain the confidence of the customers by providing them a quality product.

#### **6.2 TESTING STRATEGIES**

##### **6.2.1 Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .It is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

##### **6.2.2 Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is

specifically aimed at exposing the problems that arise from the combination of components.

### **6.2.3 Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identifying Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

### **6.2.4 System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

### **6.2.5 White Box Testing**

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It has a purpose. It is used to test areas that cannot be reached from a black box level.

### **6.2.6 Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a test in which the software under test is treated as a black box .You cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

### **6.2.7 Quality Testing**

Quality assurance is a way to avoid mistakes in the project’s product or service, and thus prevent problems for your stakeholders. It’s the part of quality management that focuses on maintaining the integrity of the product or service, which gives stakeholders the confidence that their quality requirements will be met.

### **6.2.8 Scalability Testing**

Scalability testing is a non-functional methodology in which an application’s performance is measured in terms of its ability to scale up or scale down the number of user requests or other such performance measure attributes. Scalability testing can be performed at hardware, software or database level.

### **6.2.9 Source Code Testing**

This examines the logic of the system. If we are getting the output that is required by the user, then we can say that the logic is perfect.

### **6.2.10 Recovery Testing**

Recovery Testing is a system that forces the software to fail in a variety of ways and verifies that the recovery is properly performed. If recovery is automatic, re-initialization, and data recovery are each evaluated for correctness.

## **Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

### **Test objectives**

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

### **Features to be tested**

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

### **Integration Testing:**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

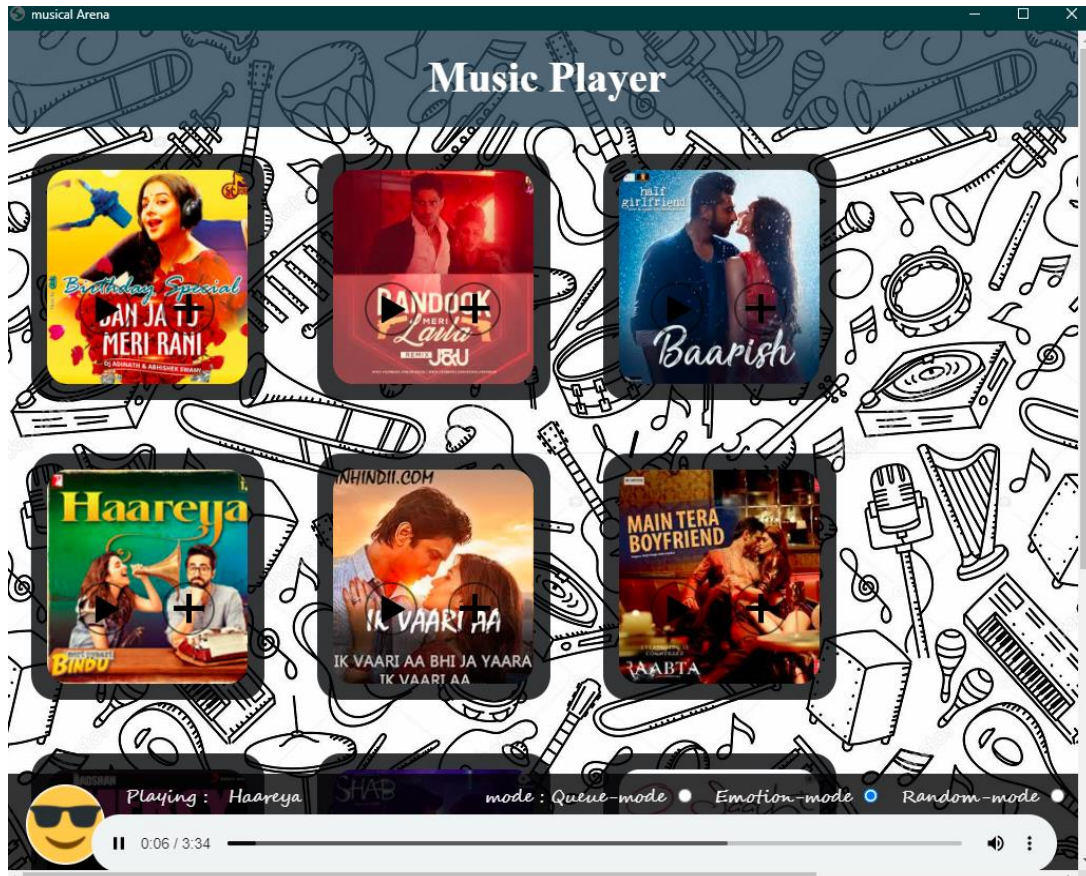
### **Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

## **6.3        SNAPSHOTS OF TEST CASES**

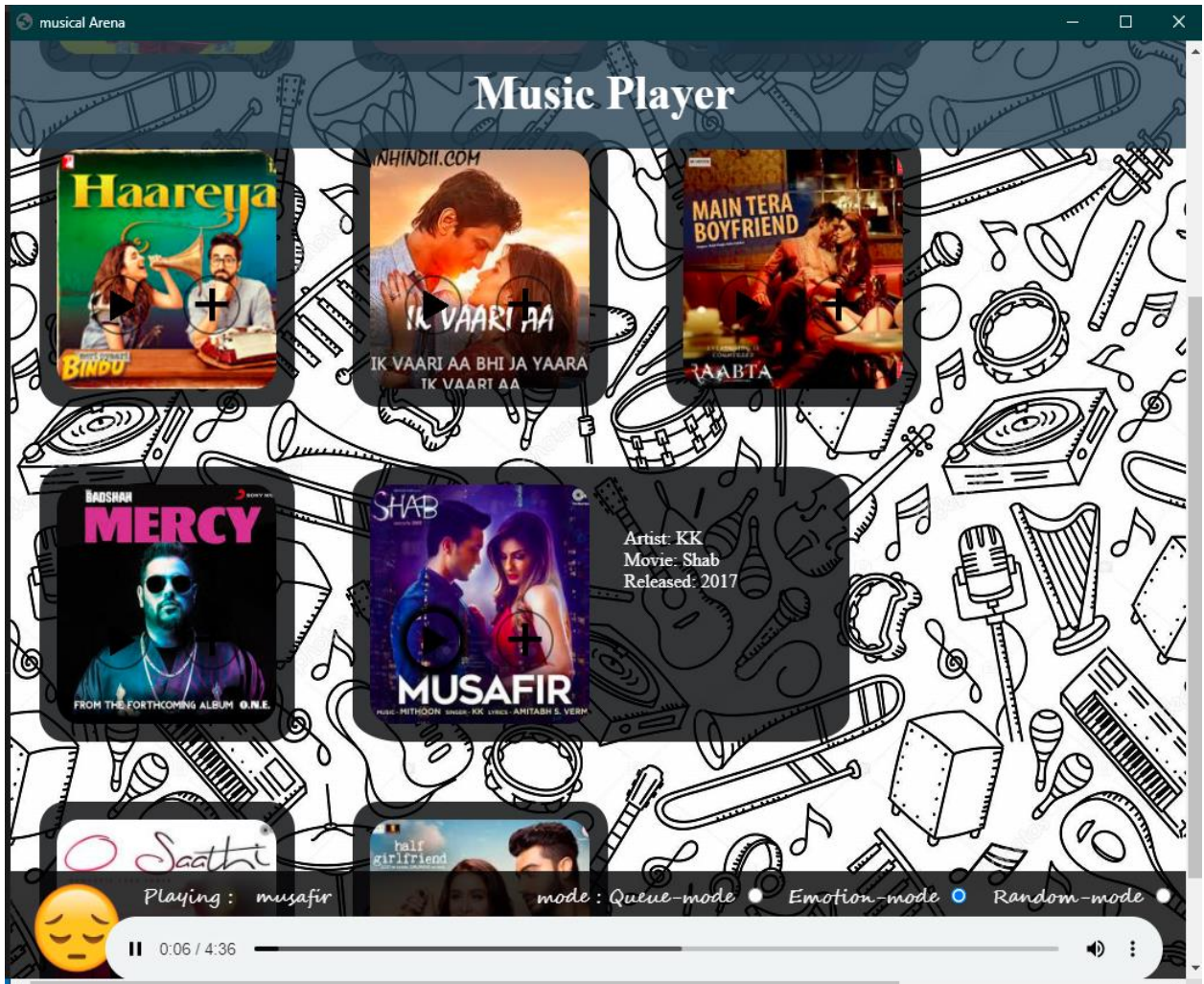
The key purpose of a **test case** is to ensure if different features within an application are working as expected. It helps testers validate if the software is free of defects and if it is working as per the expectations of the end users. **Test cases** ensure good test coverage.

**TESTCASE-1** : If the user expresses happy emotion, happy songs are played with displaying happy emoji in the audio player.

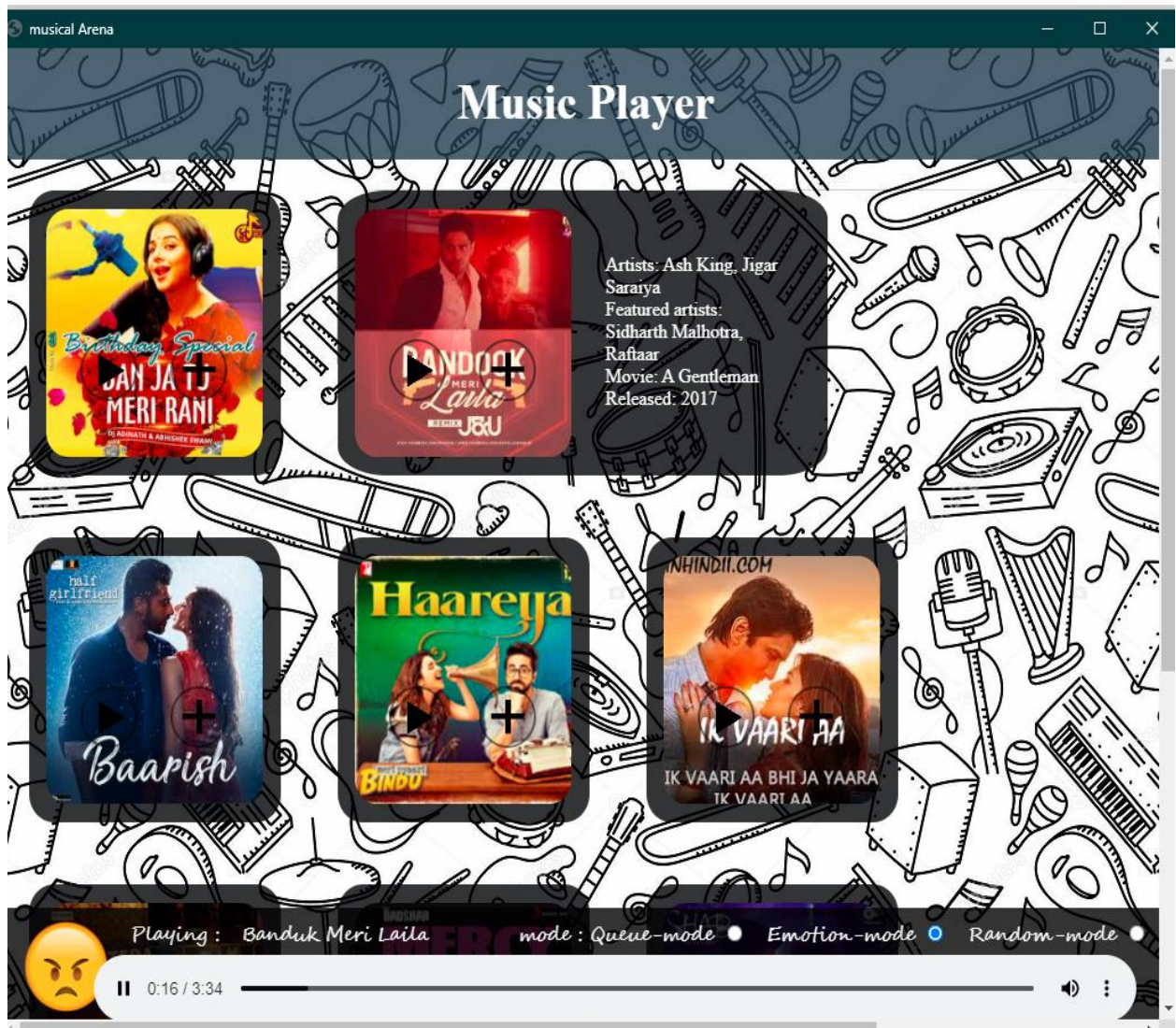




**TEST CASE-2 :** If the user expresses sad emotion, sad songs are played with displaying sad emoji in the audio player.

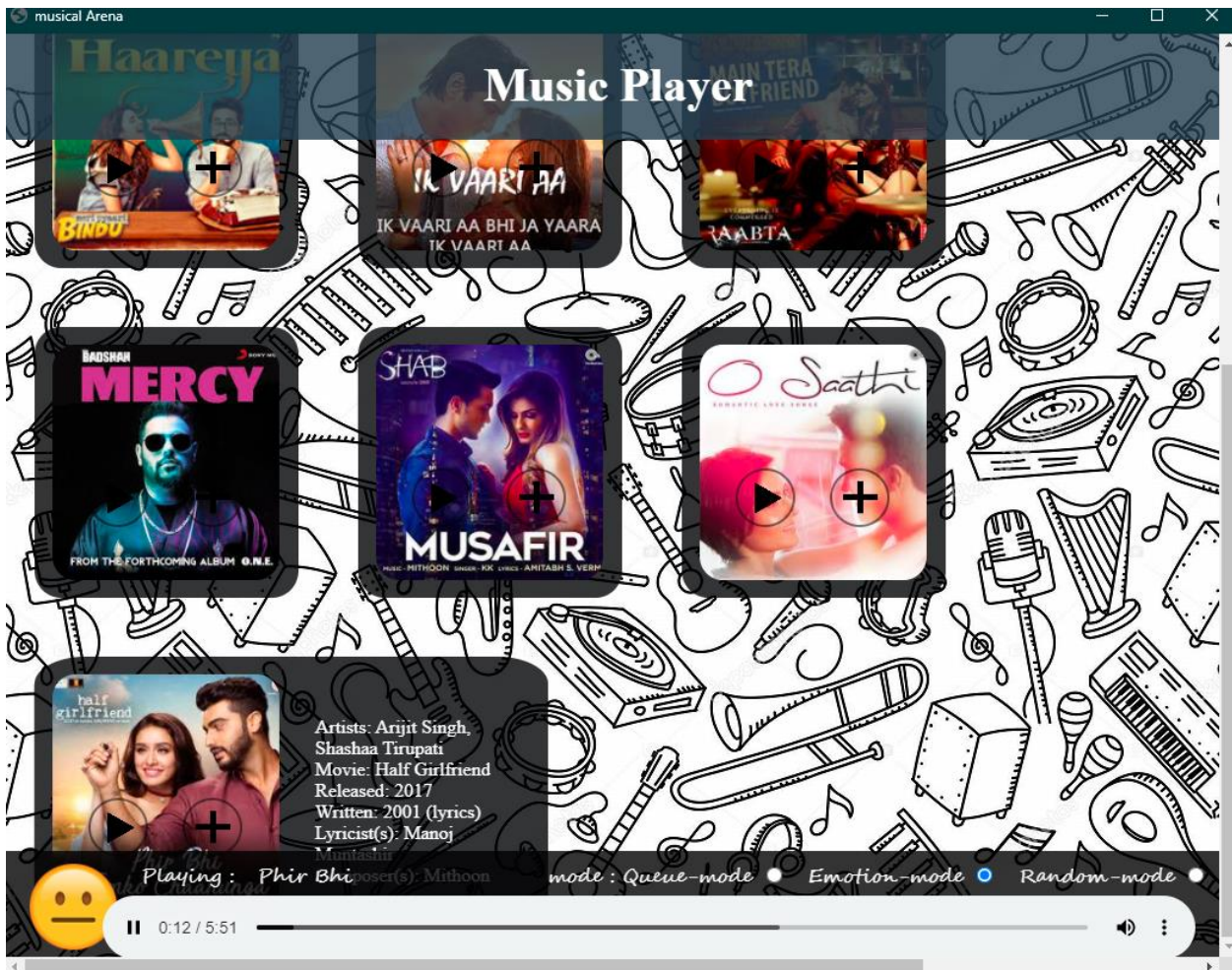


**TEST CASE-3 :** If the user expresses angry emotion, angry songs are played with displaying angry emoji in the audio player.





**TEST CASE-4 :** If the user expresses neutral emotion, neutral songs are played with displaying neutral emoji in the audio player.



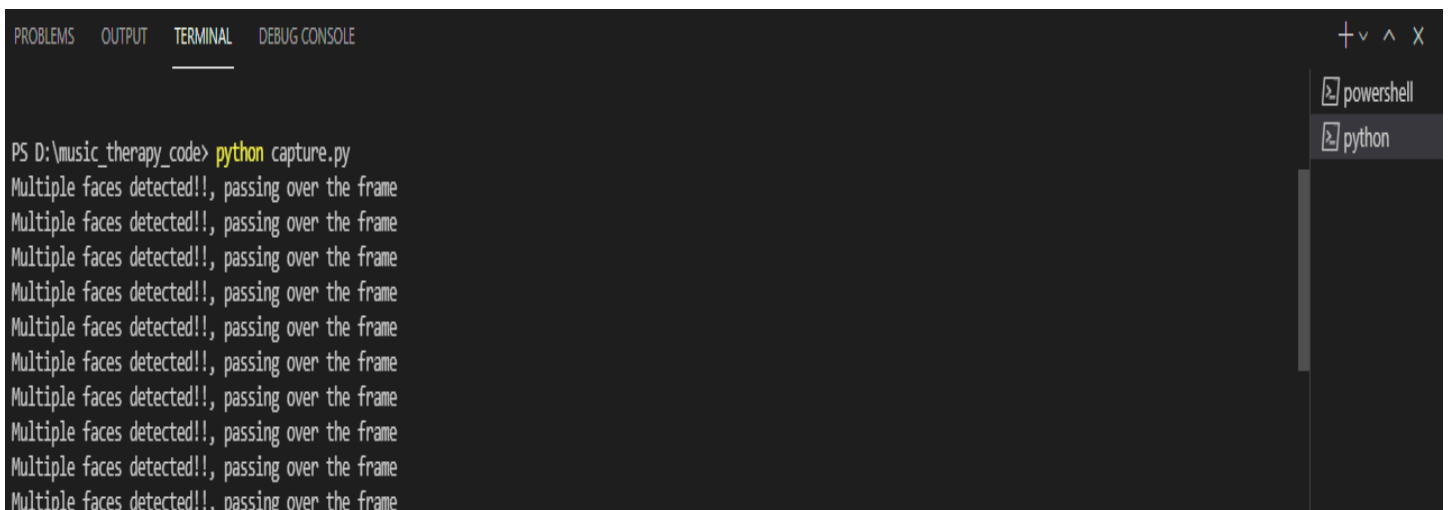
**TESTCASE-5** : If there is no face detected by the camera the output is displayed as below.



The screenshot shows a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'TERMINAL', and 'DEBUG CONSOLE', with 'TERMINAL' being the active tab. In the top right corner, there is a window title bar that says 'powershell' with standard window control icons. The terminal content shows a command prompt 'PS D:\music\_therapy\_code>' followed by the command 'python capture.py'. Below the command, there are ten lines of output, each stating 'No faces detected!!, passing over the frame'.

```
PS D:\music_therapy_code> python capture.py
No faces detected!!, passing over the frame
No faces detected!!, passing over the frame
No faces detected!!, passing over the frame
No faces detected!!, passing over the frame
No faces detected!!, passing over the frame
No faces detected!!, passing over the frame
No faces detected!!, passing over the frame
No faces detected!!, passing over the frame
No faces detected!!, passing over the frame
No faces detected!!, passing over the frame
```

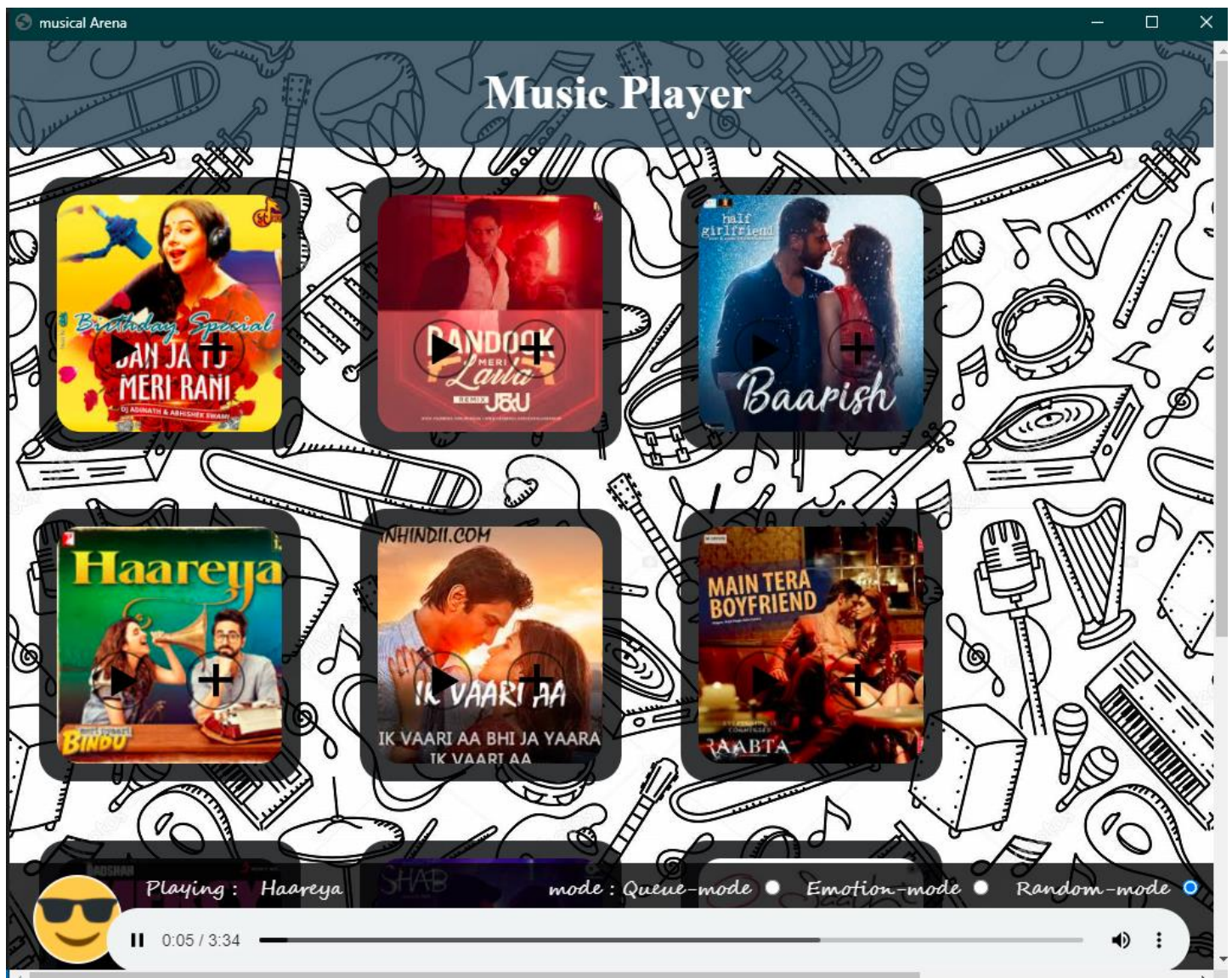
**TEST CASE-6** : If multiple faces are detected by the camera the output is displayed as below.



The screenshot shows a terminal window similar to the one above. The 'TERMINAL' tab is active. The window title bar now shows 'python' instead of 'powershell'. The terminal content shows the same command prompt and command 'python capture.py'. However, the output consists of ten lines, each stating 'Multiple faces detected!!, passing over the frame'.

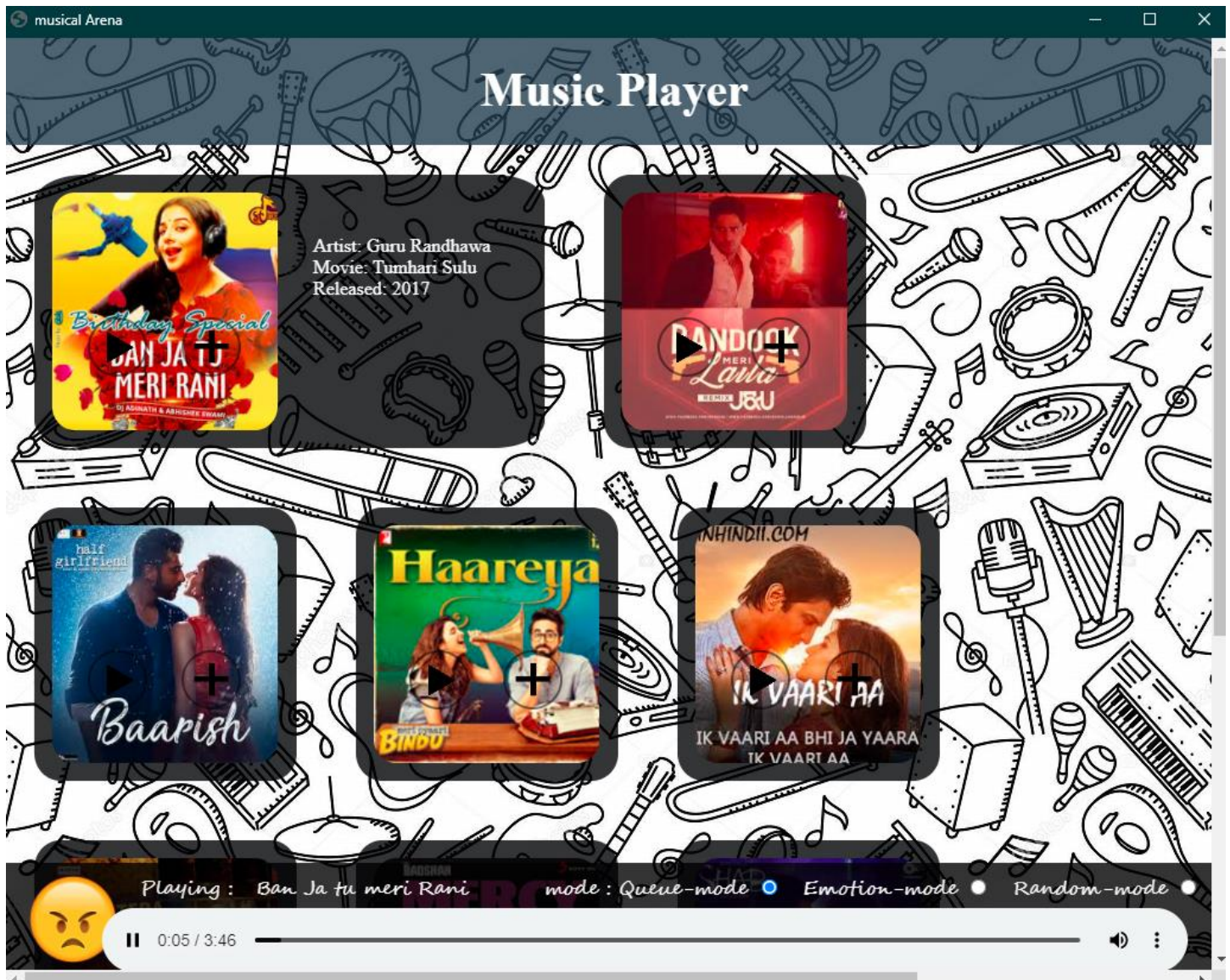
```
PS D:\music_therapy_code> python capture.py
Multiple faces detected!!, passing over the frame
Multiple faces detected!!, passing over the frame
Multiple faces detected!!, passing over the frame
Multiple faces detected!!, passing over the frame
Multiple faces detected!!, passing over the frame
Multiple faces detected!!, passing over the frame
Multiple faces detected!!, passing over the frame
Multiple faces detected!!, passing over the frame
Multiple faces detected!!, passing over the frame
Multiple faces detected!!, passing over the frame
```

**TESTCASE-7 :** If we select the random mode in the music player ,the songs will be played in random mode.





**TESTCASE-8:** Queue mode is the default mode which will be enabled after running the implementation code. In queue mode songs are played according to the songs in the audio player.



## **CHAPTER 7**

### **CONCLUSION**

#### **7.1 CONCLUSION**

Emotion recognition is a rising trend which helps in reducing the time involved in completing tasks. Selecting songs based on emotion is a complex task. So, this system helps the user by playing songs based on the emotion of the user. Emotion of the user can be recognized using any of the three features. Using any one of the features, users can listen to the songs based on the emotion. This project has been developed to give us great advancement in the field of machine learning technology. This system fulfils to sort out the music based on the emotions of the user such as whether it is happy or sad. So, totally our work aims to develop a player which is based on user need and it helps to revive in case of free time or leisure time if we want to hear music based on our current situation.

#### **7.2 FUTURE ENHANCEMENTS**

- Image capturing can be made more efficient in low light environments.
- More accurate playlist can be generated
- Even more compact device can be designed
- To include other emotions.

## CHAPTER 8

### APPENDICES

#### 8.1 SAMPLE CODE SNIPPETS

```
import numpy as np
import glob
import random
import cv2

fishface=cv2.face.FisherFaceRecognizer_create()
data={}

def update(emotions):
    run_recognizer(emotions)
    print("Saving model...")
    fishface.save("model2.xml")
    print("Model saved!!")
```

```
def make_sets(emotions):
    training_data=[]
    training_label=[]

    for emotion in emotions:
        training=sorted(glob.glob("dataset/%s/*" %emotion))
        for item in training:
            gray=cv2.imread(item,0)
            #gray=cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
            training_data.append(gray)
            training_label.append(emotions.index(emotion))
    return training_data, training_label

def run_recognizer(emotions):
    training_data, training_label=make_sets(emotions)
    print("Training model...")
    print("The size of the dataset is "+str(len(training_data))+ " images")
    fishface.train(training_data, np.asarray(training_label))
```



## 8.2 BIBLIOGRAPHY

1. Ramya Ramanathan, Radha Kumaran, Ramrohan, Rajat Gupta, Vishalakshi Prabhu “An Intelligent Music Player Base On Emotion Recognition” Published By IEEE International Conference On Computational Systems And Information Technology For Sustainable Solutions 2017 Page no 299-303.
2. Pratik Gala, Raj Shah, Vinee Shah, Yash Shah, Mrs. Sarika Rane “MoodyPlayer: A Music Player Based On Facial Expression Recognition” Published by IRJET Volume: 05 Issue: 04 on April-2018 ISSN: 2395-0056.
3. F. ADAT, C. MAAOUI and A. PRUSKI “Human –Computer Interaction Using Emotion Recognition From Facial Expression” Published by IEEE UKSim 5th European Symposium on Computer Modeling And Simulation in 2011.
4. Kyogu Lee and Minsu Cho “Mood Classification from Musical Audio Using User Group-Dependent Model” Published by IEEE 10th International Conference on Machine Learning and Application on 2011 (978-0- 7695-4607-0/11).
5. JungHyun Kim, Seungjae Lee, WonYoung Yoo “Implementation And Analysis Of Mood-Based Music Recommendation System” Published by ICACT on 2013 ISBN 978-89-968650-1-8.
6. Sushmita G. Kamble and Asso. Prof. A.H. Kulkarni “Facial Expression Based Music Player” Published by IEEE International Conference on Advance Computing, ICACCI on Sep, 2016.
7. Karthik Subramanian Nathan, Manasi Arun and Megala S kannan “EMOSIC-An Emotion Based Music Player for Android” Published by IEEE, ISSPIT in 2017.
8. Shlok Gilda, Hussain Zafar, Chintan Soni and Kshitija Waghrekar “Smart Music Player Integration Facial Emotion Recognition And Music Mood Recommendation” Published by IEEE WISPNET on 2017.
9. Renuka S. Deshmukh, Vandana Jagtap and Shilpa Paygude “Facial Emotion Recognition system Through Machine Learning” Published by ICICCS in 2017.
10. Swathi Swaminathan and E. Glenn Schellenberg “Current Emotion Research In Music Psychology” Published Emotion Review Vol. 7 ,No.2 on April 2015 (189-197), ISSN 1754-0739.
11. Maruthi Raja S K, Kumaran V, Keerthi Vasan Aand Kavitha N “Real Time Intelligent Emotional Music Player Using Android” Published by Journal for Research Volume 03, Issue 01 on March 2017.
12. Prof. Vijaykumar R. Ghule, Abhijeet B. Benke , Shubham S. jadhav and Swapnil A. Joshi “Emotion Based Music Player Using Facial Recognition” Published by IJRCCF Volume 05, Issue 02 on Feb 2017.