Final Report on

Feelio

Submitted to

SPL II Evaluation Committee 2020-2021
Bachelor of Science in Software Engineering
Institute of Information Technology
Noakhali Science and Technology University

Submitted By

Kazi Ashikur Rahman
ashik2515@student.nstu.edu.bd

Md Mamun Hossain
mamun2515@student.nstu.edu.bd

Md Foysal Mahmud
foysal2515@student.nstu.edu.bd

Supervised By

Rafid Mostafiz
Lecturer
Institute of Information Technology
rafid.iit@nstu.edu.bd

Submission Date: 17th September, 2023

Final Report on Feelio

By

Kazi Ashikur Rahman
MUH2025008M
Year-03 Term-01
ashik2515@student.nstu.edu.bd

Md Mamun Hossain
MUH2025019M
Year-03 Term-01
mamun2515@student.nstu.edu.bd

Md Foysal Mahmud
MUH2025025M
Year-03 Term-01
foysal2515@student.nstu.edu.bd

Approved By

Rafid Mostafiz
Lecturer
Institute of Information Technology
raifid.iit@nstu.edu.bd

Table of Contents

List of Figures

List of Tables

1. Project Description

1.1 Introduction

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, abbreviations, references and overview of the SRS. The aim of this document is to gather and analyze and give an in-depth insight of the complete **"Feelio".** By presenting the problem statement in detail, the purpose of this document is to collect, evaluate, and provide a deeper understanding of the whole "**Fellio**" application. Nevertheless, it also concentrates on the capabilities required by stakeholders and their needs. The detailed requirements of the **"Feelio"** are provided in this document.

1.1.1 Motivation

Our project is driven by the need to transform the way people experience music. We believe that music's profound impact on emotions deserves a tailored and therapeutic approach. Traditional methods of music selection often fall short, leading to frustration and missed emotional connections. By creating an Emotion-Based Music Recommendation System, we aim to provide users with personalized music suggestions aligned with their current emotional state. We're motivated to break the monotony of manual music selection, inject innovation into the process, and enhance the connection between emotions and music. Our goal is to bring joy, relaxation, and therapeutic benefits to users, ultimately improving their overall well-being and creating a new era of personalized music experiences.

1.1.2 Objectives

- **General Objective**

The main objective of this project is to develop the "Emotion Based Music Player" for all kinds of music lovers which aimed to serve as a platform to assist individuals to play and listen to the songs according to his emotions. It is aimed to provide a better enjoyment of entertainment to the music lovers.

- **Specific Objective**

The Specific Objective for this project is specified as below:

i. To propose a facial expression detection model to detect and analyze the emotion of an individual.
ii. To accurately detect the four basic emotions, namely normal, happy, surprise, fearful, disgusted, angry.

iii. To integrate the music player into the proposed model to play the music based on the emotions detected.

1.2 Target Users

- Customers who loves music.
- *End-users who actually use the product directly or indirectly. For our system student, teacher, people who loves music.*

1.3 Requirements

We have collected these requirements from our SRS document.

Table 1: List of requirements

No.	Requirement	Completed
FR-1	User registration and login a registered account of Feelio.	YES
FR-2	Emotion detection from image	YES
FR-3	Emotion detection from real time video	YES
FR-4	Emotion detection from text	YES
FR-5	Classification of emotions	YES
FR-6	Music playlist generation	YES
FR-7	Admin control	YES
FR-8	Logout from the system	YES
PR-1	Emotion detection with high accuracy	YES

1.4 Models, Tools and Resources

1.4.1 Model

SPL I has had a significant influence on our project. As we iteratively developed our project using the methodology we had chosen, we encountered a variety of issues. We want to implement something new this time, and after learning from the project from last year, we have decided to use the agile model to do it. Break tasks into smaller iterations using the agile model. The project risk is reduced and the overall project delivery time requirements are reduced due to the project's breakdown into smaller components.

1.4.2 Tools and Resources

Table 2: List of tools

Category	Name or Description
Text Editor	Visual Studio Code
Framework	Django
Server	Server of Django
RDBMS	SQL
Language	HTML, CSS, JavaScript, Python
Learning Resource	1. Head First HTML and CSS: A Learner's Guide to Creating Standards-Based Web Pages 2nd Edition by Elisabeth Robson 2. Head First JavaScript Programming: A Brain-Friendly Guide 1st Edition by Eric Freeman 3. Python: Head First Python by Paul Barry

1.5 Project Members

Team Members	Supervisor
Kazi Ashikur Rahman (MUH2025008M) Md Mamun Hossain (MUH2025019M) Md Foysal Mahmud (MUH2025025M)	Rafid Mostafiz Lecturer Institute of Information Technology

2. User Guide

2.1 User Side

Login

This is the login page. If the user is registered then he can login into our system by providing a username and password. If the user is new to our system then he must click on the "Create one" button.

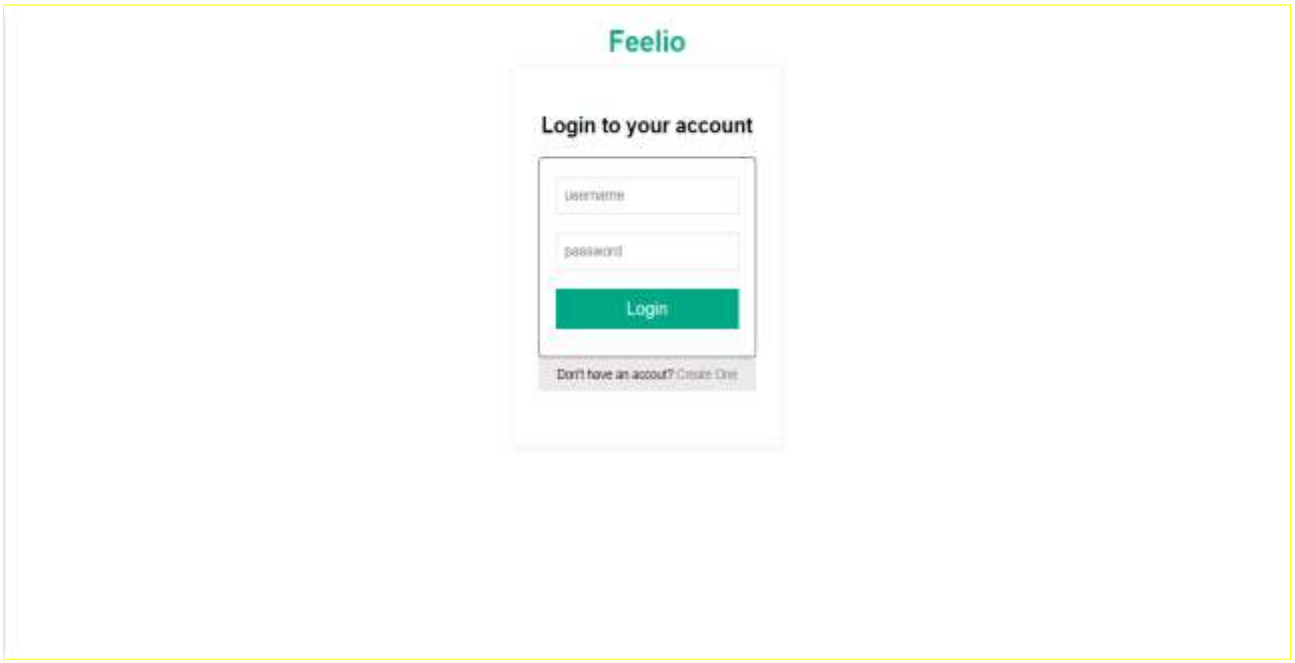

Figure 1: Login

Create Account / Registration

If you want to log in to our system then you have to register by clicking the "create one" button on the login page. After that, you have to provide the required information and then click the "Registration" button. If you already have an account then click on "Login" button.

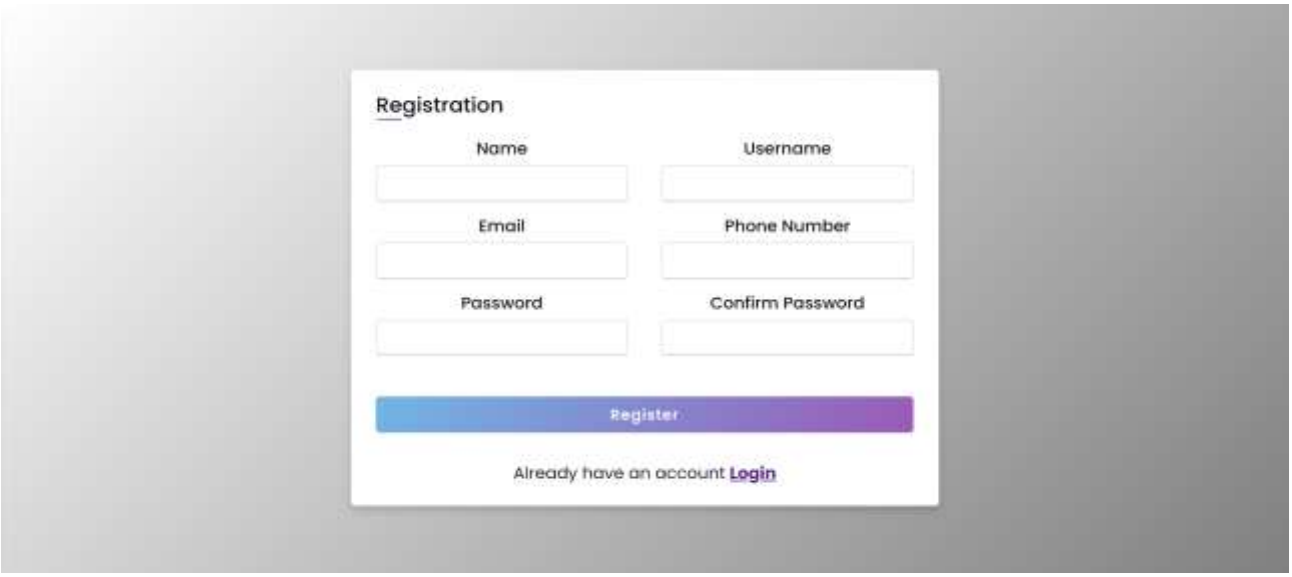

Figure 2: Create Account/ Registration

Verification

After clicking on the "Registration" button on the registration page the system sends a verification link by email when the user clicks on the link the account will be activated.

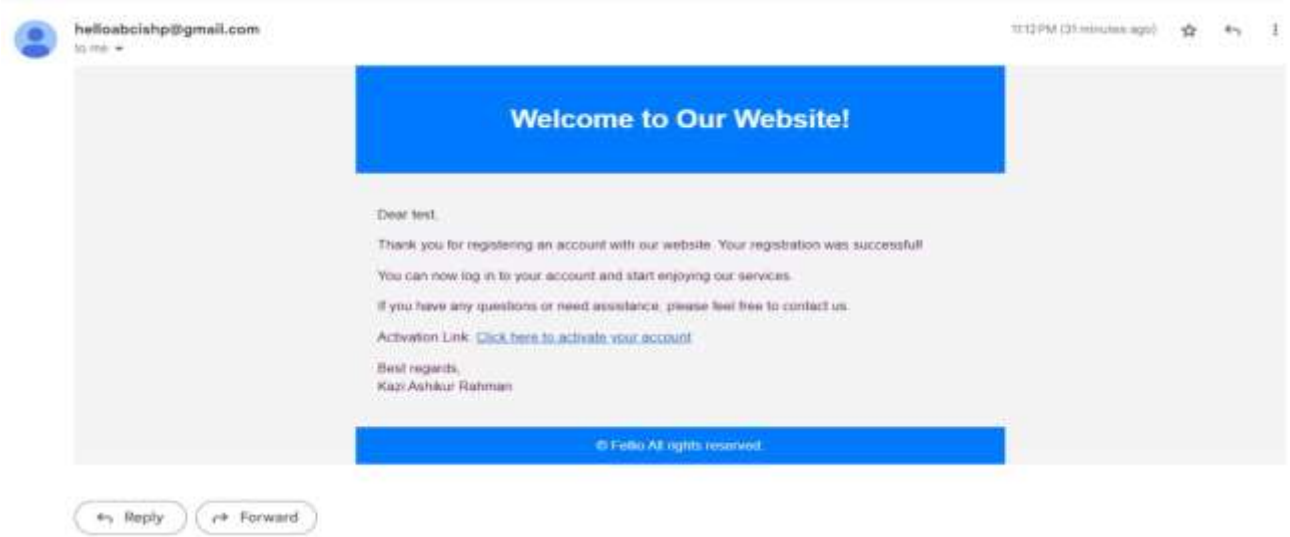

Figure 3: Verification

ord?" option then you will be able to

Figure 4: Forgotten Password

Home page

This is the home page of our system. When a user logs into their account he will go to the homepage. Here user can perform many tasks if he clicks "Capture Video" then he will go to emotion detection from the video page, if he clicks "Enter Text" then he will go to emotion detection from text, if he clicks "Select Image" then he will go to emotion detection image page. Users can logout from this page if he/she wants.

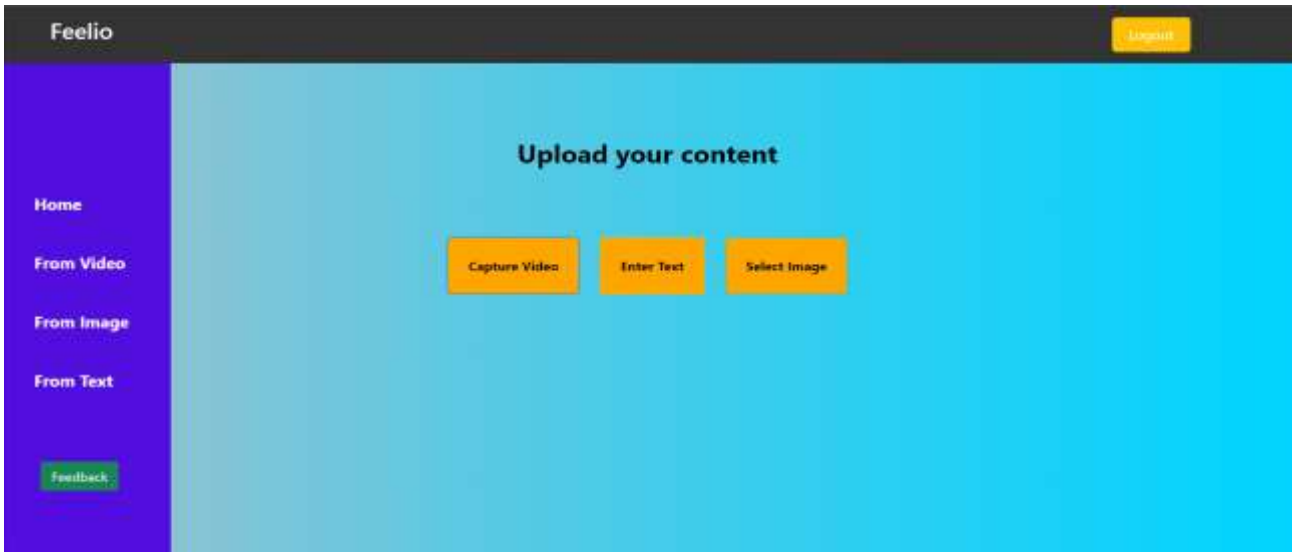

Figure 5: Home Page

Emotion detection from video

On this page, emotion will be detected from real-time video streaming. After detecting emotion, the playlist of music will be shown beside the camera interface. Every song has the song name, Album name, and Artist name. If the user clicks on the song's name then we will go to the music on YouTube then he/she can listen to it.

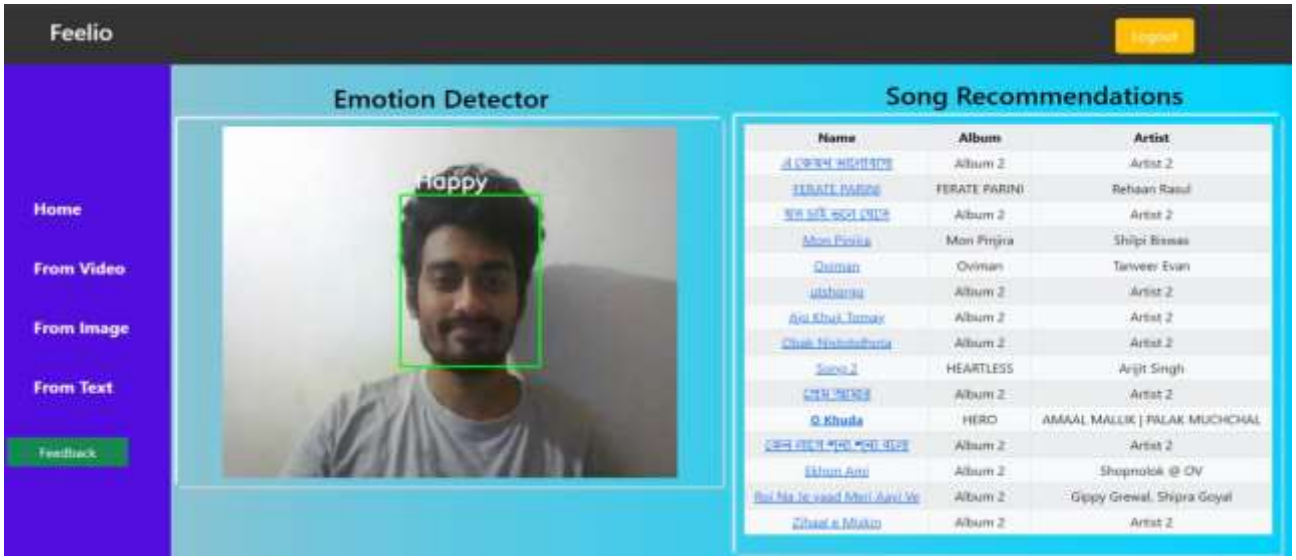

Figure 6: Emotion detection from video

Figure 7: Emotion detection from video

Emotion detection from image

On this page, emotion will be detected from uploaded. The user must select an image from the file and then click "Recognize Emotion". After detecting emotion, the playlist of music will be shown. Every song has the song name, Album name, and Artist name. If the user clicks on the song's name then we will go to the music on YouTube then he/she can listen to it.

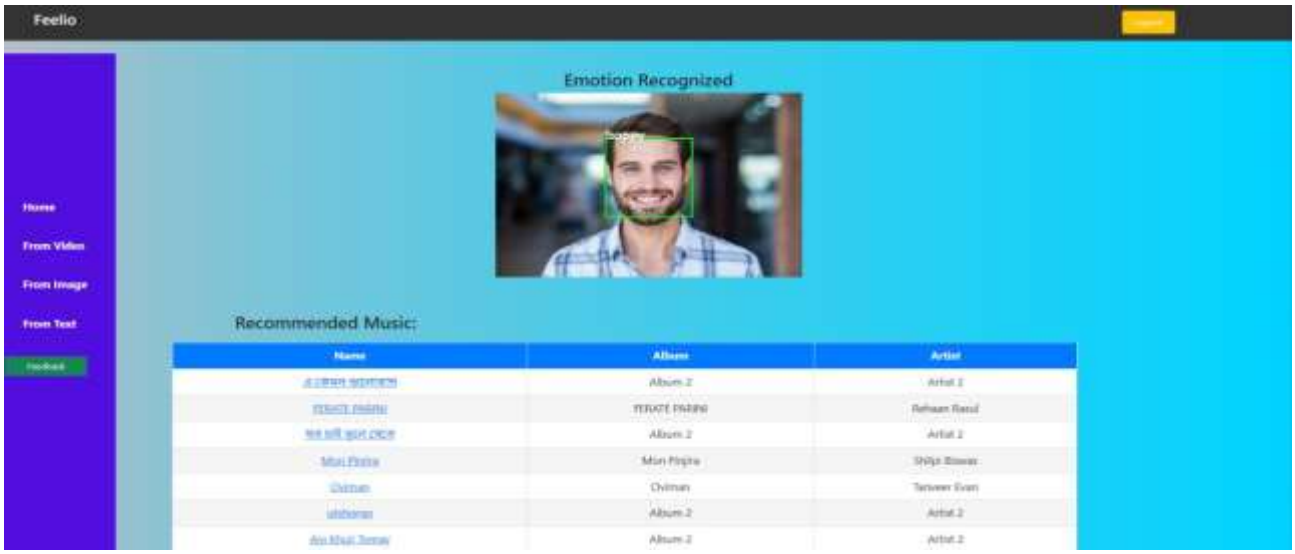

Figure 8: Emotion detection from image

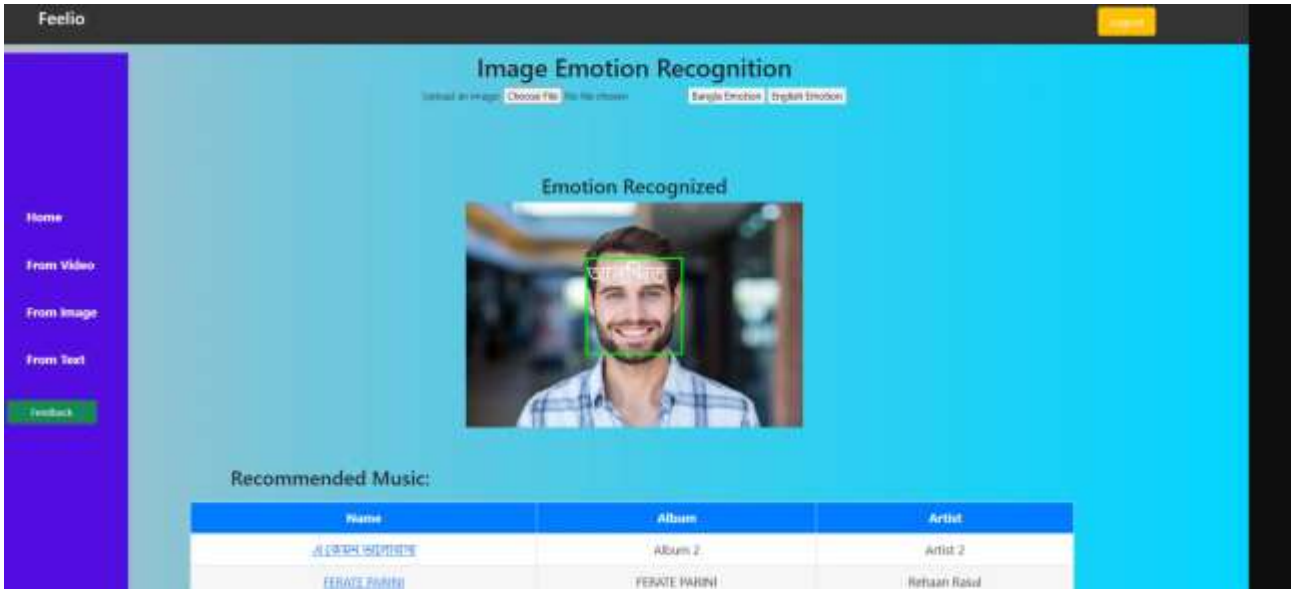

Figure 9: Emotion detection from image

Emotion detection from text

On this page, emotion will be detected from the text. The user must type their text in the selected area and then click on "Submit". After detecting emotion, the playlist of music will be shown. Every song has the song name, Album name, and Artist name. If the user clicks on the song's name then we will go to the music on YouTube then he/she can listen to it.

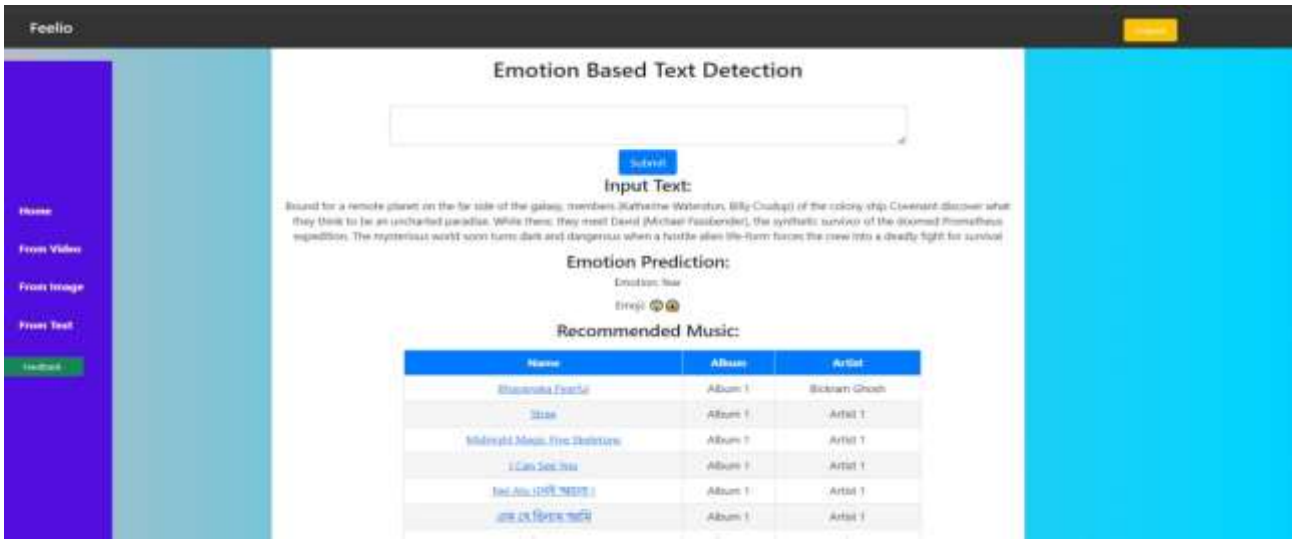

Figure 10 : Emotion detection from text

Feedback

User can provide their valuable feedback about our project

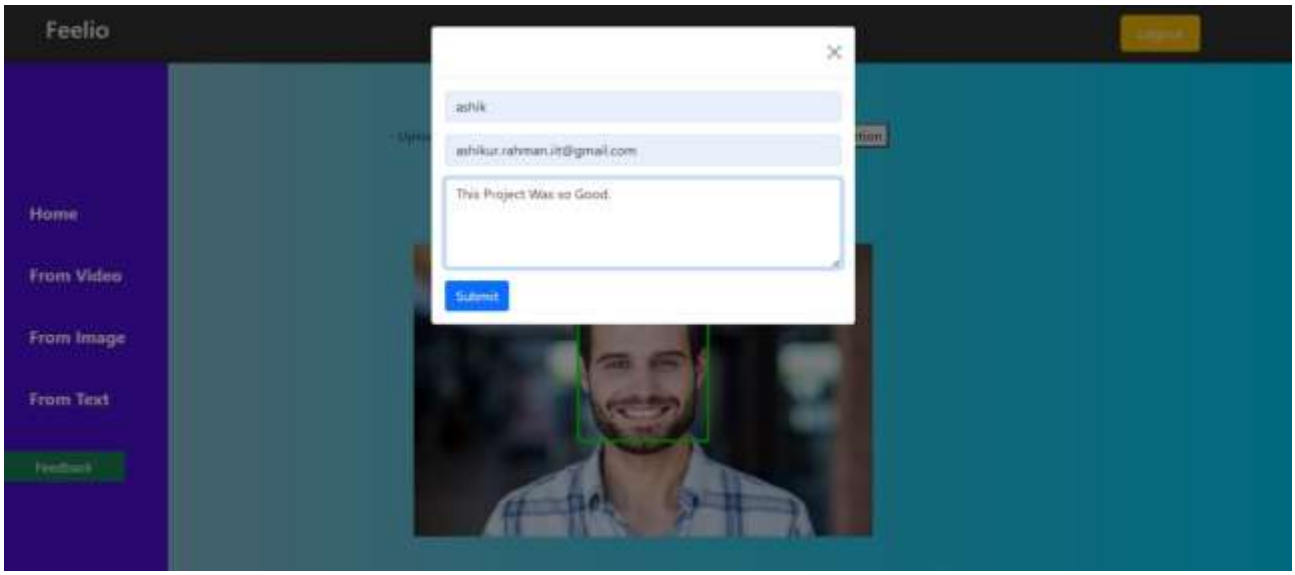

Figure 11 : Feedback

2.2 Admin Side

Login

This is the login page for admin. If the user is registered then he can login into our system by providing a username and password. If the username or password is not valid then must register by command.

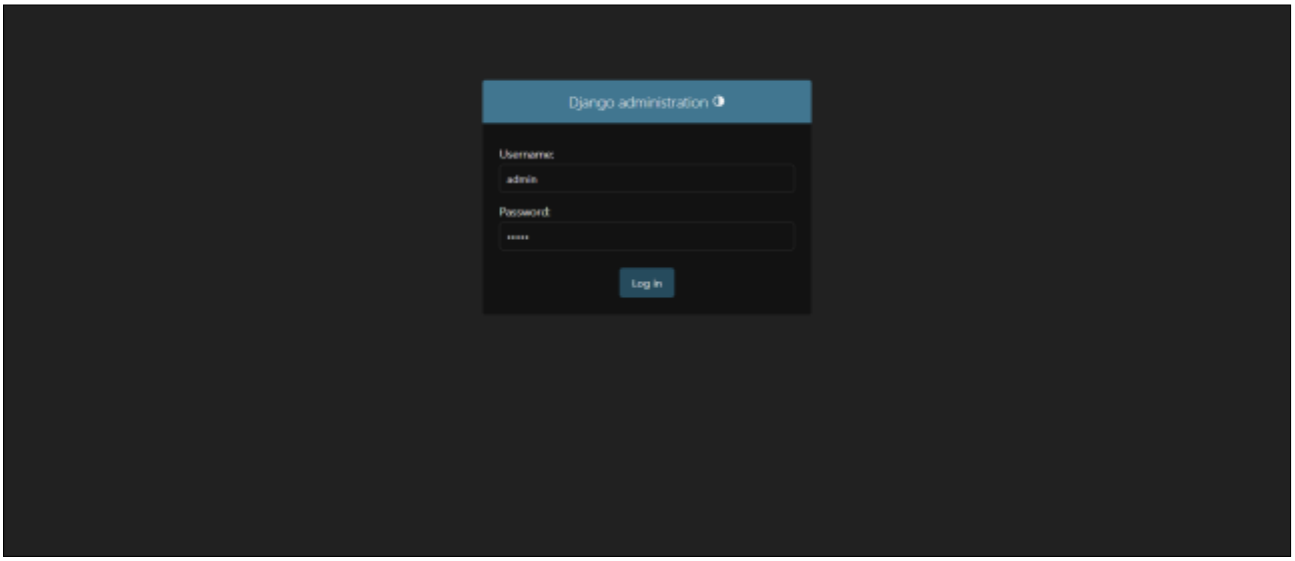

Figure 12: Login

Home Page

This is the homepage for admin. Admin will be able to see all tables that are created. If he wants to change password then must click on "CHANGE PASSWORD" If he wants to logout then he must click on "Logout".

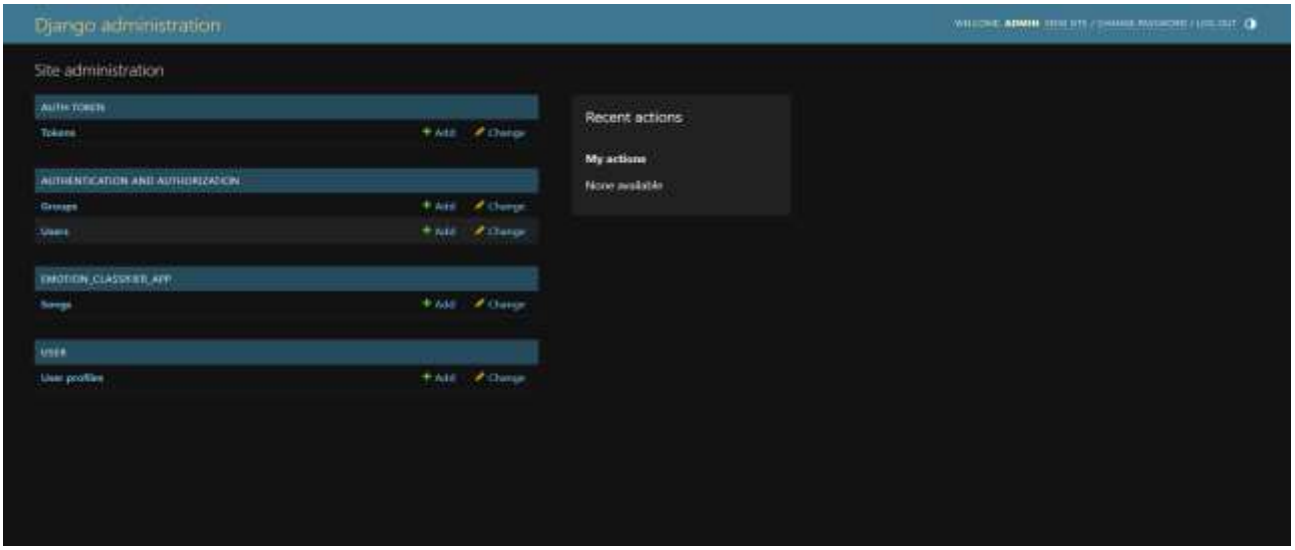

Figure 13: Home Page

User information

Here the admin can see all the information of the user which was given during the registration. Admin can add new user by clicking "ADD USER"

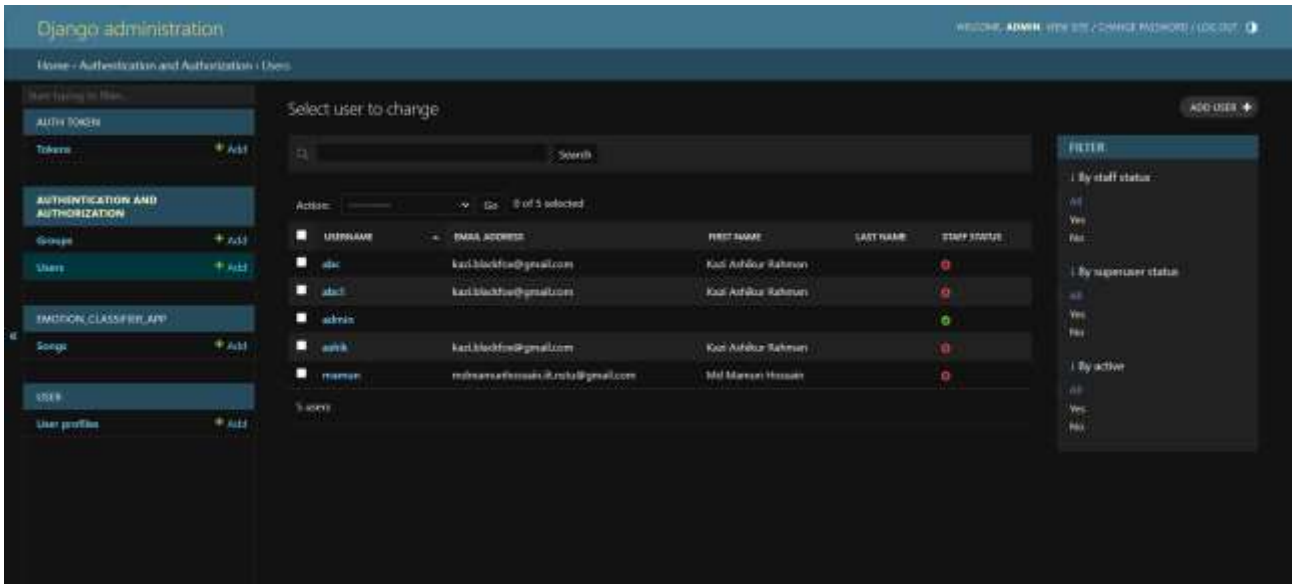

Figure 14 : User Information

Add user

User can add new user by providing required information and clicking on "Save" button

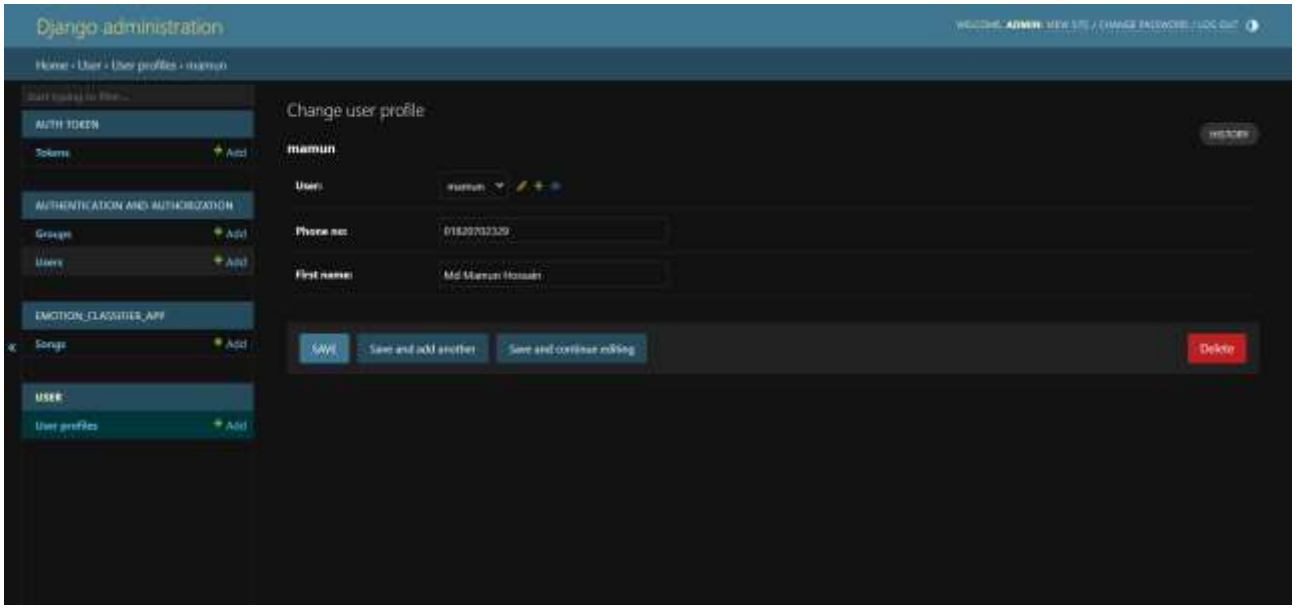

Figure 15 : Add User

Song Information

Here all information about song will be shown. If admin want to add a new song then he/she must on clicks on "ADD SONG" button.

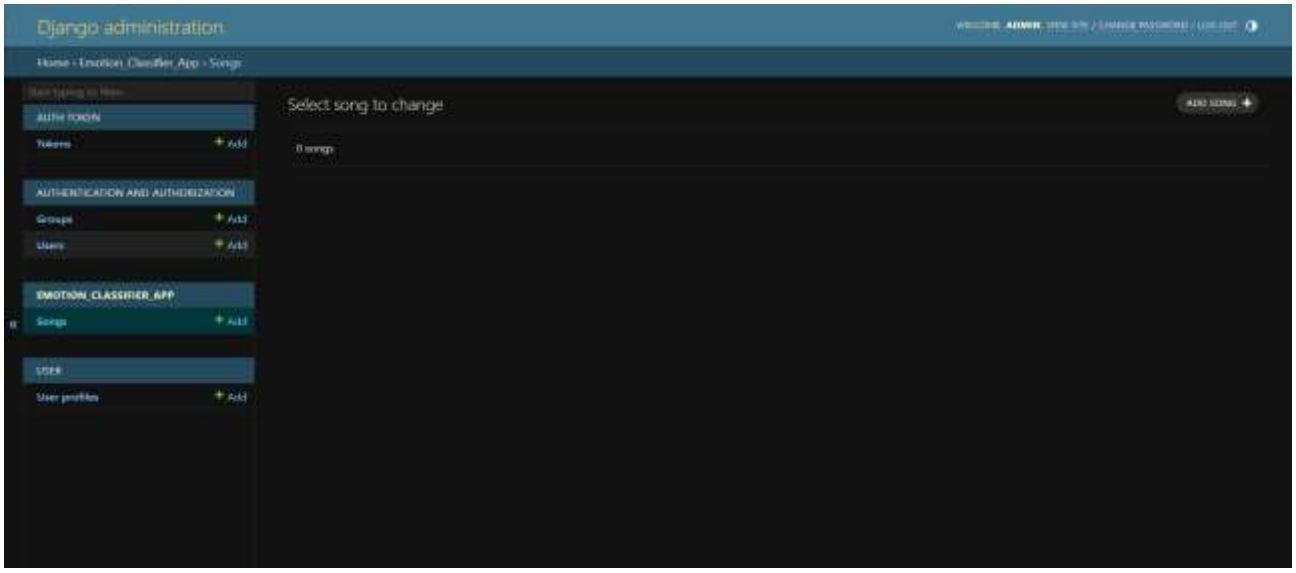

Figure 16 : Song Information

Add new song

Admin can add a new song by providing required information such as emotion, Name, Link etc. Finally clicks on "SAVE" button.

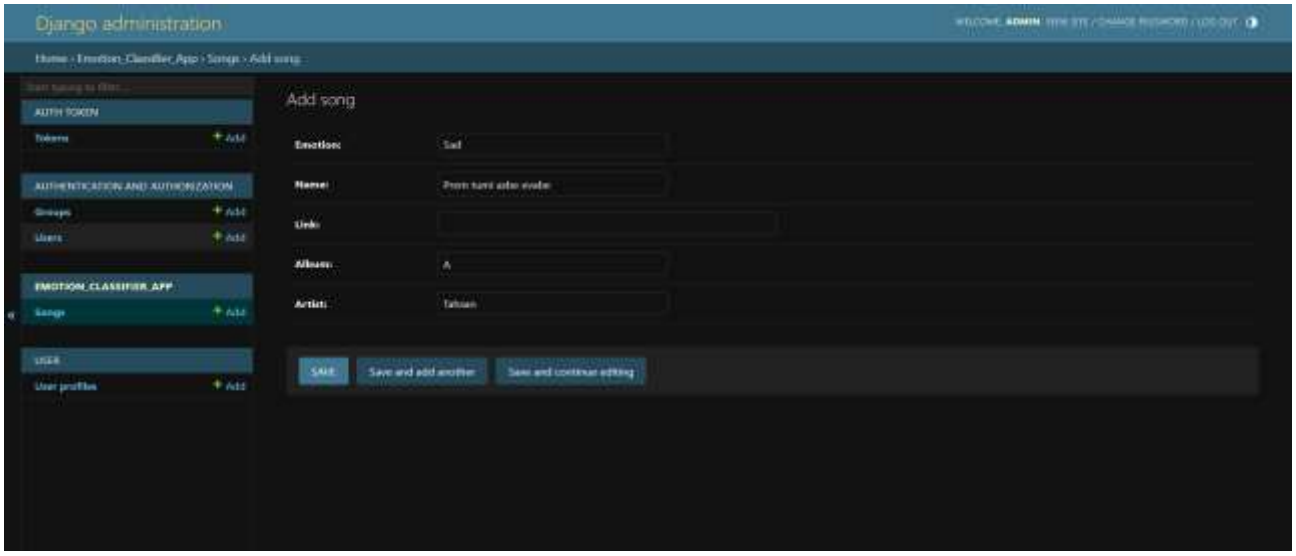

Figure 17 : Add new song

Change password

Admin can change their password. First, he needs to provide the old password, set a new password with a maintained set of rules , confirm the password, and finally click on the "CHANGE PASSWORD" button.

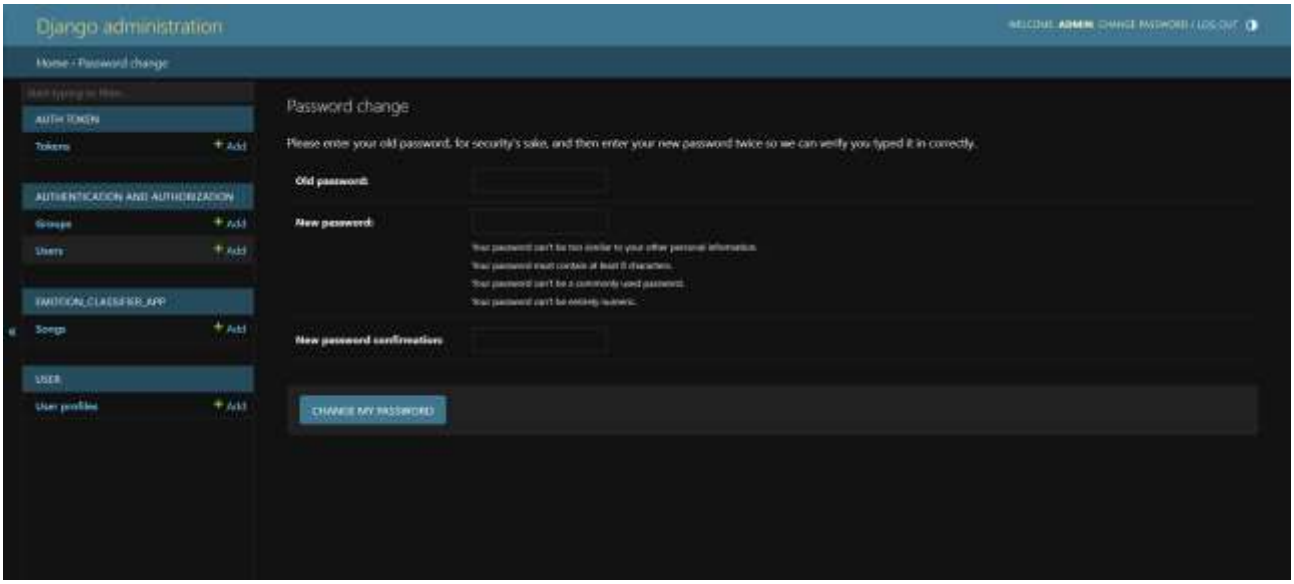

Figure 18 : Change Password

Logout

Admin can logout from the system by clicking on "Logout" button After logging out, if he wants to login to the system again he must click on the "Login" button.

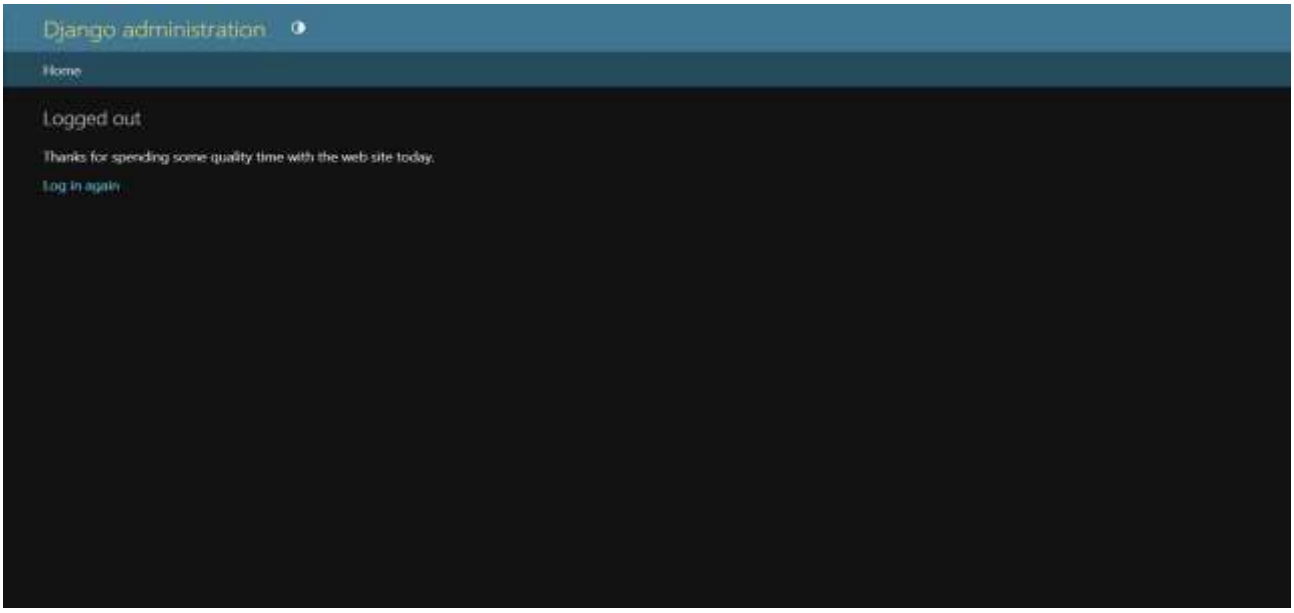

Figure 19 : Logout

3. Source Code Documentation

3.1 Emotion_classifier_project

3.1.1 managy.py

This is the default django's file that helps to run Django project in any browser.

3.1.2 db.sqlite3

This is default database created by Django framework.

3.1.3 bulk.py

This file facilitates so that we can insert data into database.

3.1.4 emotion_classifier_project

This folder contains so many folders and files such as

3.1.4.1 settings.py

The settings.py file is a crucial part of the project's configuration. It is typically located within the project's main directory and contains various settings and configurations for the Django project. Here are some of the key aspects and settings you might find in a typical settings.py file.

3.1.4.2 urls.py

The urls.py file is used to define the URL patterns and routing for your project. This file is crucial for mapping URLs to views, which are responsible for handling HTTP requests and generating responses

3.1.5 User

3.1.5.1 admin.py

The admin.py file is used to configure and customize the Django admin interface

3.1.5.2 apps.py

The apps.py file is part of an app's configuration and is used to provide metadata and configuration for the Django app

3.1.5.3 forms.py

The forms.py file is used to define and manage forms. Forms are a fundamental part of web applications, allowing users to input data, submit it to the server, and interact with the application.

3.1.5.4 models.py

The models.py file is a fundamental component used to define the structure and schema of your application's database tables. It is part of Django's Object-Relational Mapping (ORM) system and follows the model-view-controller (MVC) architectural pattern.

3.1.5.5 Template

Html files stored in here.

3.1.5.6 Static

All CSS, JavaScript, images stored in here

3.1.5.7 migrations

In Django, migrations is a management command used to generate database migration files based on changes you've made to your application's models. Migrations are a way to keep track of database schema changes and apply those changes to the database.

3.1.6 Models

Machine learning trained models like .h5 ,json , .xml are stored in here.

3.1.7 emotion_clasifier_app

3.1.7.1 views.py

The views.py file is where you define the view functions or classes that handle HTTP requests and return HTTP responses. Views are responsible for processing user input, interacting with models and databases, and rendering templates to generate the final HTML pages that are sent to the client's web browser

3.1.7.2 urls.py

The urls.py file is used to define the URL patterns and routing for your project. This file is crucial for mapping URLs to views, which are responsible for handling HTTP requests and generating responses.

3.1.7.3 models.py

The models.py file is a fundamental component used to define the structure and schema of your application's database tables. It is part of Django's Object-Relational Mapping (ORM) system and follows the model-view-controller (MVC) architectural pattern.

3.1.7.4 camera.py

Capture the real time videos in the concept of machine learning

3.1.7.5 apps.py

The apps.py file is part of an app's configuration and is used to provide metadata and configuration for the Django app

3.1.7.6 admin.py

The admin.py file is used to configure and customize the Django admin interface. The Django admin interface is a powerful tool that allows you to manage and interact with your application's data through a web-based interface.

3.1.7.7 Template

Html files stored in here.

3.1.7.8 Static

All CSS, JavaScript, images stored in here

3.1.7.9 migrations

In Django, migrations is a management command used to generate database migration files based on changes you've made to your application's models. Migrations are a way to keep track of database schema changes and apply those changes to the database.

3.2 Lines of Code

Table 03 List of lines of code

SL	Class	LOC	NCLOC	CLOC	Density of Comments
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
	Total Project				

4. Challenges and Future Work

The main challenges we faced during developing this system are,

1. **Emotion Detection Accuracy:** Ensuring accurate and real-time emotion detection from users can be challenging, as emotions are complex and can vary greatly from person to person. Improving the system's ability to accurately interpret emotional cues is essential.
2. **Diverse Music Catalogs**: Handling a vast and diverse catalog of music with different genres, moods. The system needs to effectively categorize and recommend songs that cater to individual preferences.
3. **Privacy Concerns**: Collecting and analyzing user emotional data can raise privacy concerns.
4. **Integration:** Ensuring the system seamlessly integrates with various music streaming platforms and devices, as well as keeping up with evolving technologies, is essential for widespread adoption.

The future work we are proposing,

1. **Enhanced Emotion Detection**: Continuously improving the accuracy of emotion detection through advanced AI and machine learning techniques will be a priority. This may involve incorporating user feedback to refine the emotional analysis algorithms.
2. **Ethical Considerations:** Continuously addressing ethical concerns regarding user data privacy and ensuring transparency in how user data is used and stored.

5. SRS Document

Requirements

User registration and login

FR-1	User Registration and Login to a registered account.		
Description	User should register his/her account for the first time and be able to log in to the account which was registered once. Already registered users will not face this stage.		
Stakeholders	System User	**Priority**	High

Emotion detection from image

FR-2	Emotion detection from uploaded images		
Description	The system shall analyze uploaded images and extract emotional features to determine the user's emotion.		
Stakeholders	System User	**Priority**	High

Emotion detection from text

FR-3	Emotion detection from uploaded text		
Description	The system shall process uploaded text and employ natural language processing techniques to detect emotions.		
Stakeholders	System User	**Priority**	High

Emotion detection from real time video

FR-4	Emotion detection from real-time video streaming		
Description	The system shall utilize computer vision algorithms to analyze real-time video streams for emotions		
Stakeholders	System User	**Priority**	High

Classification of emotions

FR-5	Classification of emotions		
Description	The system shall categorize detected emotions into happy, sad, excitement, and neutral to form a basis for music recommendations.		
Stakeholders	System User	**Priority**	High

Music playlist generation

FR-6	Music playlist generation		
Description	Based on the detected emotions, the system shall generate personalized playlists containing songs matching the user's emotional state.		
Stakeholders	System User	**Priority**	High

Admin Control

FR-7	Admin Control		
Description	There will be admin panel, they can control everything after logging in admin panel. Action will be taken by admin against anyone who posts fake or bad comments.		
Stakeholders	Admin	**Priority**	Medium

Logout

FR-8	User log out from their account.		
Description	The user will be able to log out of his/her account at the end of his need. Users will need to log in again for later use.		
Stakeholders	System User	**Priority**	Medium

Password protection

DR-1	Users' password will be protected		
Description	Hashing will be used to keep users' passwords protected.		
Stakeholders	Developers	**Priority**	High

Video dataset for emotion detection

DR-3	Video dataset for emotion detection		
Description	Video dataset for emotion detection: The system shall require a labeled dataset of videos samples that express different emotional states, including happiness, sadness, excitement, and neutrality.		
Stakeholders	Data Provider, System User	**Priority**	High

Image dataset for emotion detection

DR-2	Image dataset for emotion detection		
Description	Image dataset for emotion detection: The system shall require a labeled dataset of image samples that express different emotional states, including happiness, sadness, excitement, and neutrality.		
Stakeholders	Data Provider, System User	**Priority**	High

DR-4	Text dataset for emotion detection		
Description	Text dataset for emotion detection: The system shall require a labeled dataset of text samples, such as user descriptions, captions, and comments, that express different emotional states, including happiness, sadness, excitement, and neutrality.		
Stakeholders	Data Provider, System User	**Priority**	High

Music database

DR-5	Music database		
Description	The system shall require a comprehensive and up-to-date music database that includes a wide range of songs, albums, and artists. The database will serve as the source for generating music recommendations based on the user's detected emotion. It should provide relevant metadata such as song titles, artists, genres, and audio features (e.g., tempo, mood) to enhance the recommendation process.		
Stakeholders	Music Database Provider, System User	**Priority**	High

High Accuracy

PR-1	High Accuracy		
Description	The system should accurately detect emotions from various sources for precise recommendations.		
Stakeholders	User, Data Science Team	**Priority**	High

Scalability

PR-2	**Scalability**		
Description	The system should handle a large number of users and process data efficiently.		
Stakeholders	System Administrator, User	**Priority**	High

Use Case Description

Table 4: Access Control (Account Creation)

Use Case	Access Control	
Goal	User wants to create an account in the "Feelio" System.	
Preconditions	N/A	
Success End Condition	A user account is created.	
Failed End Condition	User account is not created.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Create Account" or "Sign up" button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	User opens the "Feelio" application.
	2	User clicks the "Create Account" or "Sign up" button.
	3	User provides required information such as name, email, password etc.
	4	Server checks Gmail is already existed or not.
	5	User activated the account.
	6	Account is created.
Alternative Flows	**Step**	**Branching Action**
	2a	User have an account.
	2b	User needs to be clicked login button.
	2c	User get access of the application.
	3a	User does not provide information.
	4a	Server shows that information invalid or used before.
	4a1	User needs to change provided information.
Quality Requirements	**Step**	**Requirement**
	1	Server will respond within 3 to 5 seconds.

Table 5 : Access Control (Login)

Use Case	Access Control	
Goal	User login to "Feelio" system.	
Preconditions	User have an account.	
Success End Condition	Successfully login to "Feelio" system.	
Failed End Condition	Unable to login.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Login" Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	User opens the "Feelio" system.
	2	User provides Gmail and password.
	3	Server confirms the password with that Gmail.
	4	User successfully logs into application.
Alternative Flows	**Step**	**Branching Action**
	2a	User does not have an account.
	2a1	User clicks the "Create Account" button to create an account.
	4a	Provides the correct email and password.
Quality Requirements	**Step**	**Requirement**
	1	User can login whenever he/she want (Availability).

Table 6 : Get Video

Use Case	Get video	
Goal	User will receive a playlist based on her/his emotion from real time video streaming.	
Preconditions	User must login into application.	
Success End Condition	User receive a playlist.	
Failed End Condition	User doesn't receive any music playlist.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Video Capture" Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	User go to video capture section.
	2	System sends a notification for accessing camera.
	3	User allows the camera access.
	4	Face detector detect the face and retrieve a mood based on expression.
	5	System accesses the database and show a playlist of music based of approved mood.
	6	User can listen the music based on their emotion.
Alternative Flows	**Step**	**Branching Action**
	3a	System sends a notification for accessing camera.
Quality Requirements	**Step**	**Requirement**
	1	Accuracy rate of emotion detection will be high.
	2	Robustness

Table 7 : Upload Image

Use Case	Upload Image	
Goal	User will be able to upload the image.	
Preconditions	User must be logged in.	
Success End Condition	Image will be uploaded successfully.	
Failed End Condition	Unable to upload.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Image upload" Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	Users go to image upload section.
	2	User browse the device section and select an image file.
	3	Upload the image by clicking "upload" button.
Alternative Flows	**Step**	**Branching Action**
	2a	Selects the image file again.
Quality Requirements	**Step**	**Requirement**
		Response time will be 2 seconds.

Table 8 : Get Image

Use Case	Get Image	
Goal	User will receive a playlist based on her/his emotion from the uploaded image.	
Preconditions	User must logged in.	
Success End Condition	User receive a playlist.	
Failed End Condition	User doesn't receive any music playlist.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Image upload" Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	User clicks on "Image upload" button.
	2	Selects an image file and clicks upload.
	3	System scan the image and retrieve a mood based on expression.
	4	System accesses the database and show a playlist of music based of approved mood.
	6	User can listen the music based on their emotion.
Alternative Flows	**Step**	**Branching Action**
		N/A
Quality Requirements	**Step**	**Requirement**
	1	Accuracy rate will be high.

Table 9 : Upload Image

Use Case	Upload Text	
Goal	User will be able to upload the typing text.	
Preconditions	User must type a text.	
Success End Condition	Text will be uploaded successfully.	
Failed End Condition	Unable to upload	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Text upload" Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	Users go to text upload section.
	2	User can type the text based their own emotion.
	3	Upload the text by clicking "upload" button.
Alternative Flows	**Step**	**Branching Action**
		N/A
Quality Requirements	**Step**	**Requirement**
		Uploaded time will be minimum.

Table 10 : Get Text

Use Case	Get Text	
Goal	User will receive a playlist based on her/his emotion from the uploaded image.	
Preconditions	User must logged in.	
Success End Condition	User will receive a playlist.	
Failed End Condition	User doesn't receive any playlist from the application.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Text upload" Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	User type the text based on their current mood.
	2	Upload the written text by clicking "upload" button.
	3	System detect the text and retrieve a mood based on expression.
	4	System accesses the database and show a playlist of music based of approved mood.
	5	User will receive a playlist.
Alternative Flows	**Step**	**Branching Action**
	2a	N/A
Quality Requirements	**Step**	**Requirement**
		Detection accuracy will be high rated.

Table 11 : Retrieve Mood

Use Case	Retrieve mood	
Goal	System retrieve the mood.	
Preconditions	Start the video of upload an image or type a text.	
Success End Condition	Successfully retrieve the user mood.	
Failed End Condition	Unable to retrieve the mood.	
Primary Actors: **Secondary Actors:**	System User	
Trigger	"Get video" or "Get image" or "Get text" button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	Users start the real-time video.
	1b	Face detector detect the face.
	1c	System retrieve a mood based on their expression.
	2	User upload an image.
	2b	System scan the image and retrieve a mood.
	3	User upload a text.
	3b	Text detector detects the text and retrieve a mood.
Alternative Flows	**Step**	**Branching Action**
		N/A
Quality Requirements	**Step**	**Requirement**
		Response time will be at most 3 seconds

Table 12 : Approved mood

Use Case	Approved mood	
Goal	User mood will be approved	
Preconditions	Camera must be on or upload any image or type any text on text upload section.	
Success End Condition	Successfully approve the user mood.	
Failed End Condition	Unable to approve.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Get video" or "Get image" or "Get text" button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	Users start the real-time video.
	1b	Face detector detect the face.
	1c	System retrieve a mood based on their expression and mood will be approved.
	2	User upload an image.
	2b	System scan the image and retrieve a mood.
	2c	The system approved the mood.
	3	User upload a text.
	3b	Text detector detects the text and retrieve a mood.
	3c	The system approved the mood.
Alternative Flows	**Step**	**Branching Action**
		N/A
Quality Requirements	**Step**	**Requirement**
		Emotion detection accuracy will be high.

Table 13 : Retrieve genre

Use Case	Retrieve Genre	
Goal	System retrieve a genre of songs based on detected emotion.	
Preconditions	Needs to be accessed database.	
Success End Condition	Successfully retrieve a genre of songs.	
Failed End Condition	Failed the retrieve genre.	
Primary Actors: **Secondary Actors:**	System User	
Trigger	N/A	
Main Success Flows	**Step**	**Action**
	1	User start the video or upload an image or upload a text.
	2	System will approve a mood based on its training.
	3	System access the database and return a genre a songs based approval mood.
Alternative Flows	**Step**	**Branching Action**
		N/A
Quality Requirements	**Step**	**Requirement**
	2	There are at least 100 songs of every genre.

Table 14 : Playlist

Use Case	Playlist	
Goal	User receive a playlist.	
Preconditions	Camera must be on or upload any image or type any text on text upload section.	
Success End Condition	User receive a playlist successfully.	
Failed End Condition	Unable to receive a playlist.	
Primary Actors: **Secondary Actors:**	System User	
Trigger	N/A	
Main Success Flows	**Step**	**Action**
	1	User start the video or upload an image or upload a text.
	2	System will approve a mood based on its training.
	3	System access the database and return a genre a songs based approval mood.
	4	System provide a playlist based on retrieval genre.
Alternative Flows	**Step**	**Branching Action**
	3	N/A
Quality Requirements	**Step**	**Requirement**
		There have been 100 songs of every playlist.

Table 15 : Manage Database

Use Case	Manage Database	
Goal	System will manage the database.	
Preconditions	N/A	
Success End Condition	Database will be managed successfully.	
Failed End Condition	Database management failed.	
Primary Actors: **Secondary Actors:**	System / Admin N/A	
Trigger	N/A	
Main Success Flows	**Step**	**Action**
	1	Go the add songs section.
	1a	Provide the song details
	1b	Select a genre of songs.
	1c	Add into specific genre.
	2	Admin can also remove the song.
	2a	Select the songs that will be removed.
	2b	Delete the song from database
	3	Admin can edit the songs details.
	3a	Provide the edited details.
	3b	Clicks the "Save" button.
	4	Admin will be able to create a new playlist.
	4a	Collect the specific numbers of songs
	4b	Provide every song details
	4c	Provide genre of music and click to add.
Alternative Flows	**Step**	**Branching Action**
		N/A
Quality Requirements	**Step**	**Requirement**
	2	N/A

Table 16 : Logout

Use Case	Logout	
Goal	Logout from "Feelio" system.	
Preconditions	System updates the board whenever	
Success End Condition	Successfully logout from "Feelio" system.	
Failed End Condition	Remain in a login state.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Logout" Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	User using the "Feelio" system after login
	2	User clicks the "Logout" button.
	4	User logout successfully.
Alternative Flows	**Step**	**Branching Action**
		N/A
Quality Requirements	**Step**	**Requirement**
		N/A

6. Activity Diagram

Activity diagram (Create Account)

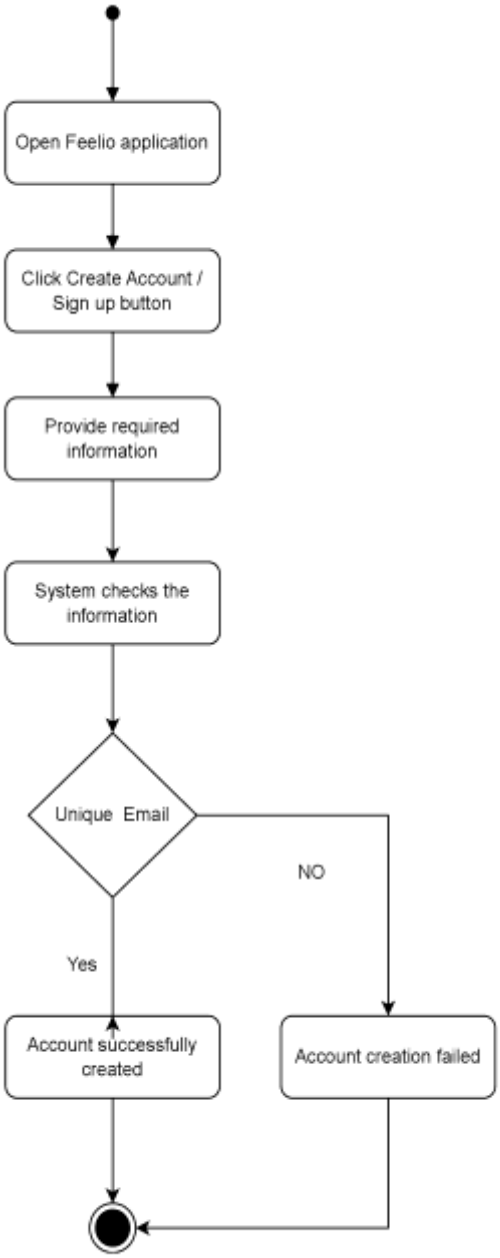

Figure 20 : Create Account

Activity diagram (Login)

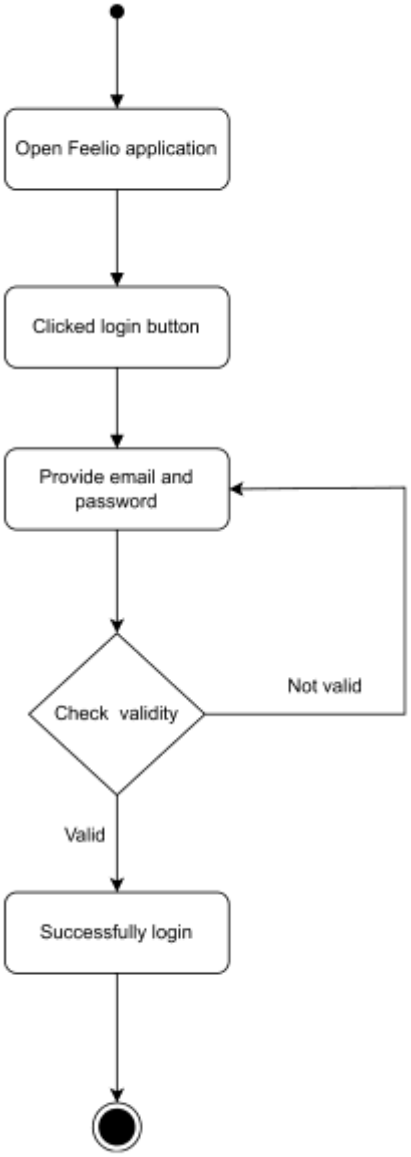

Figure 21: Login

Activity diagram (Get video)

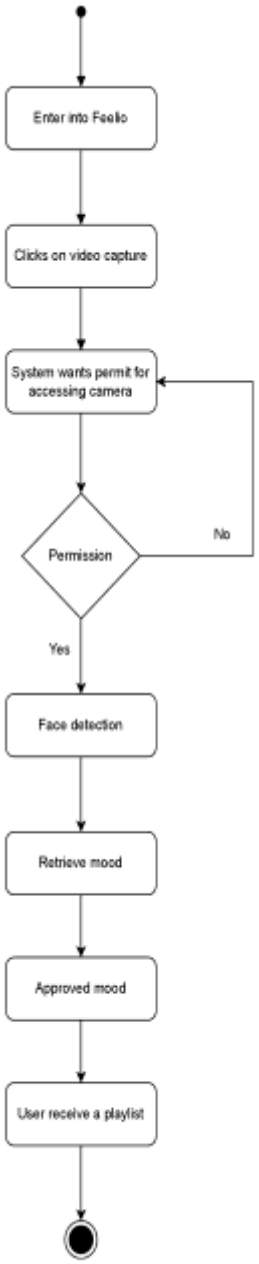

Figure 22: Get Video

Activity diagram (Image upload)

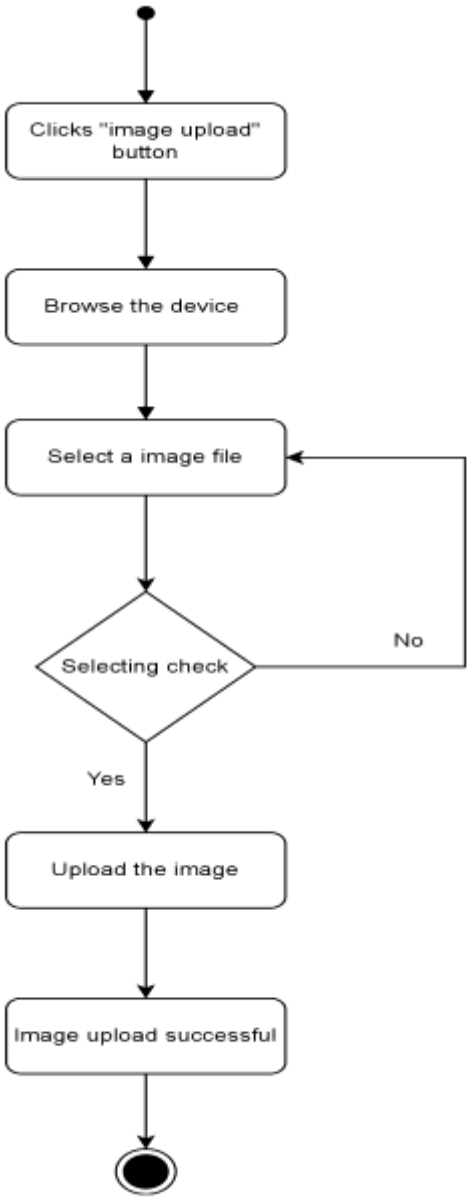

Figure 23 : Image Upload

Activity diagram (Get image)

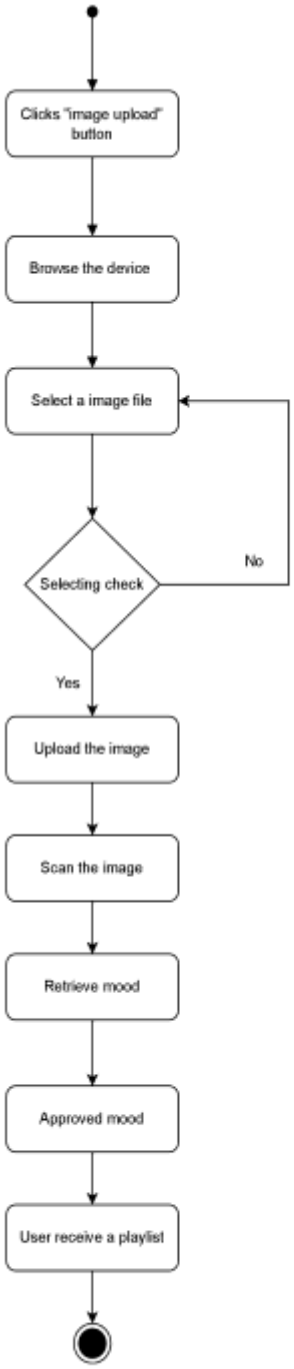

Figure 24 : Get Image

Activity diagram (Text upload)

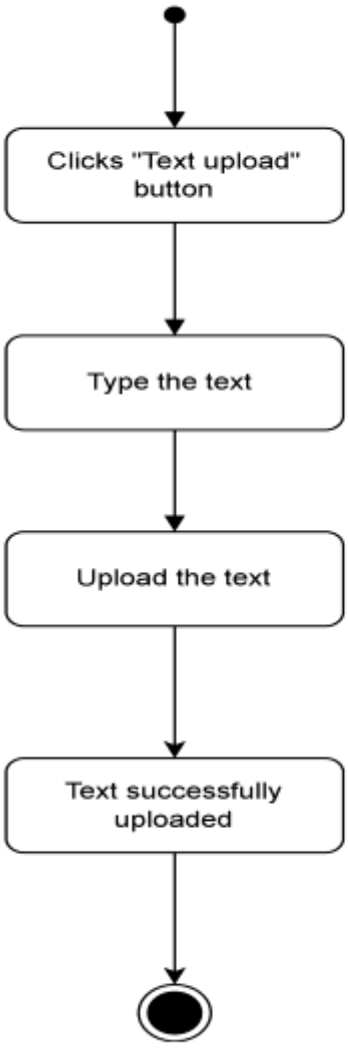

Figure 25: text Upload

Activity diagram (Get text)

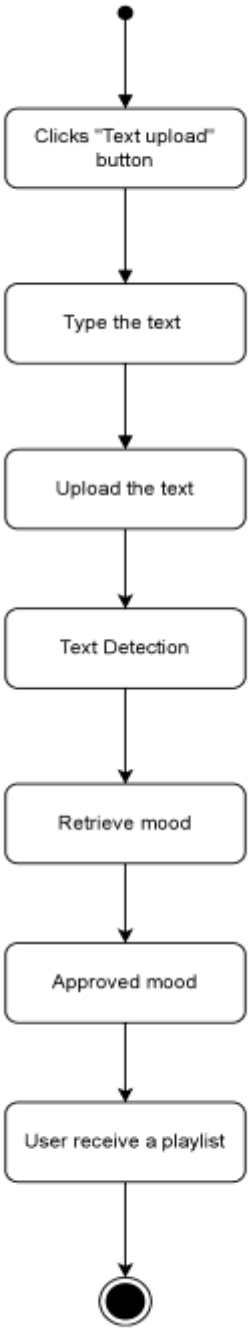

Figure 26: Get Text

Activity diagram (Retrieve mood)

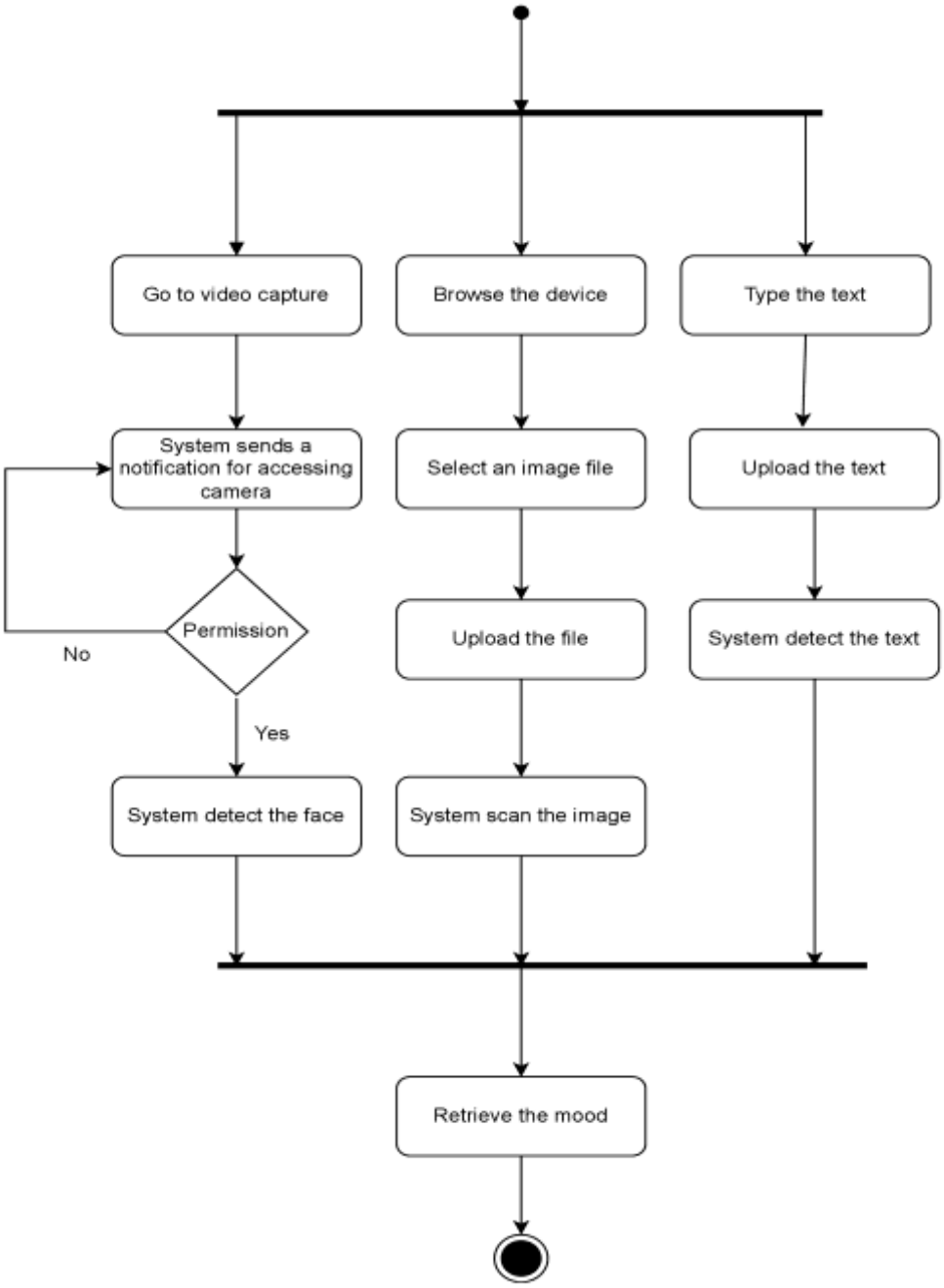

Figure 27: Retrieve Mood

Activity diagram (Approved mood)

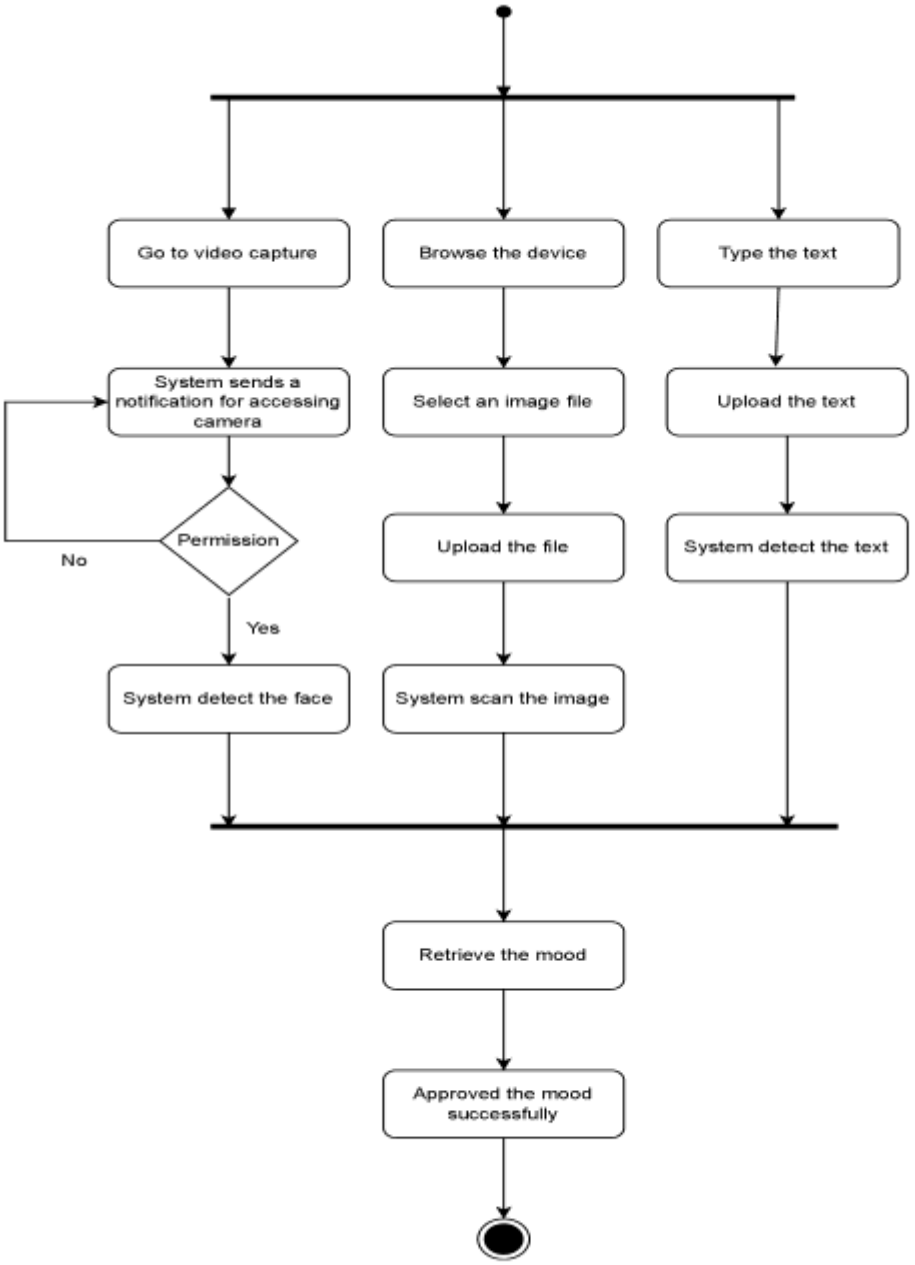

Figure 28: Approved Mood

Activity diagram (Retrieve Genre)

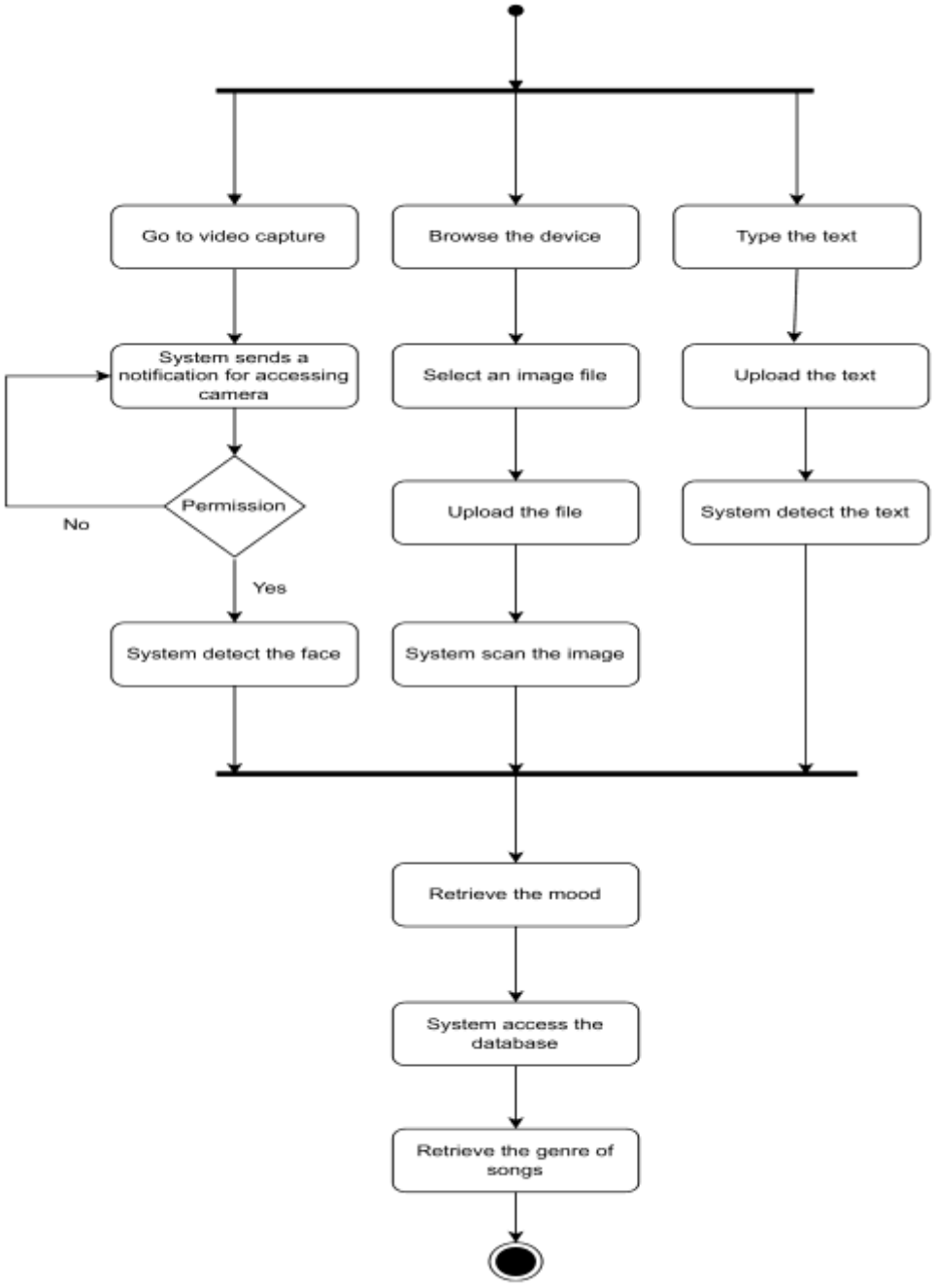

Figure 29: Retrieve Genre

Activity diagram (Playlist)

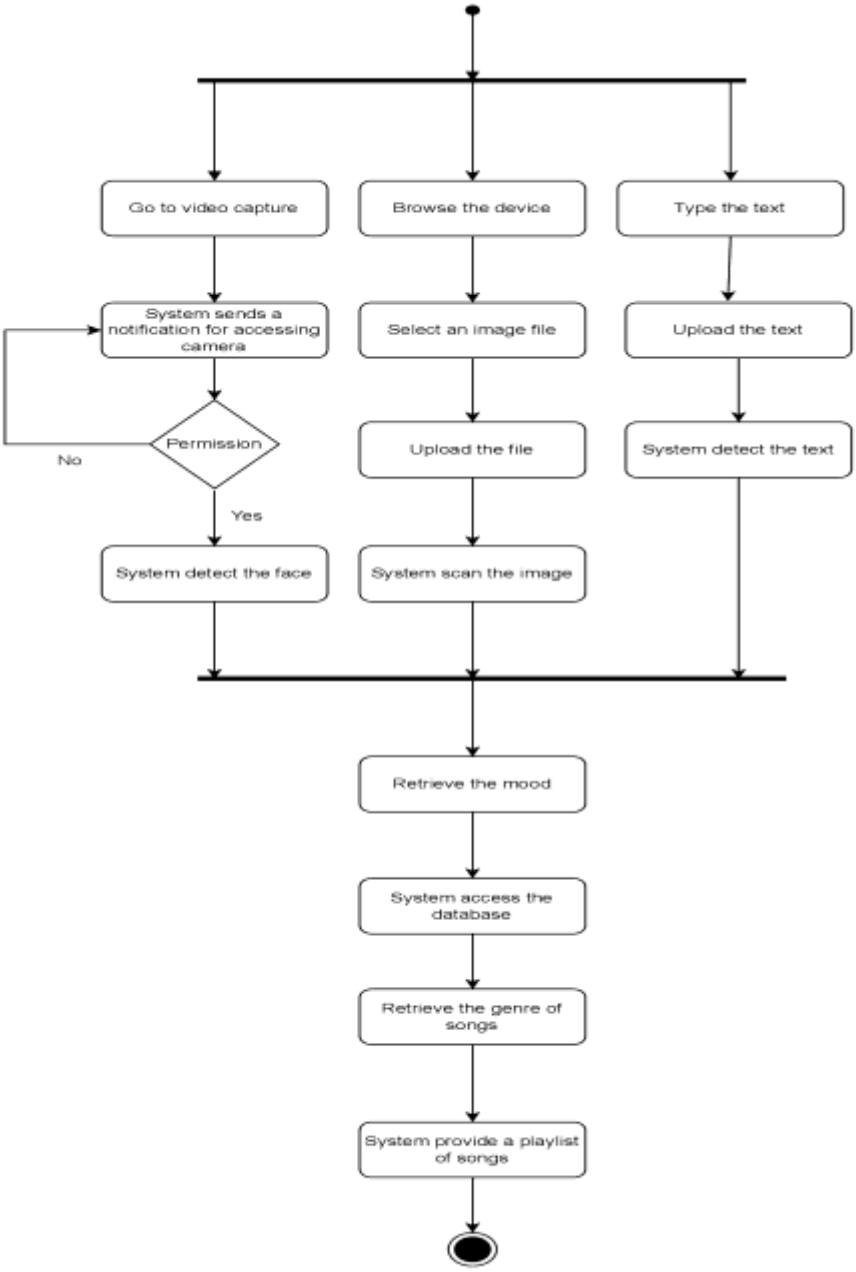

Figure 30: Playlist

Activity diagram (Manage database)

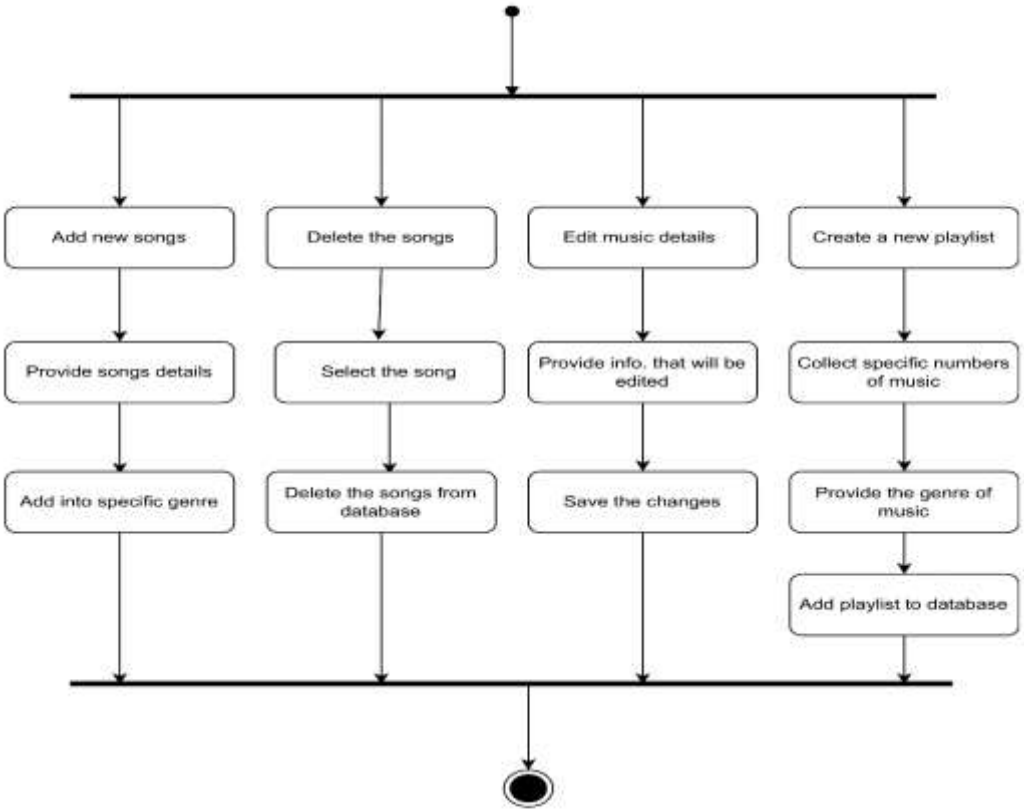

Figure 31: Manage Database

Activity diagram (Log out)

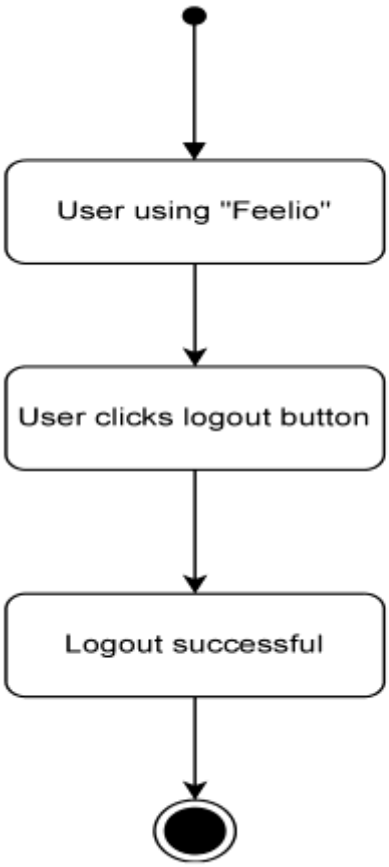

Figure 32: Logout

7. Use case Diagram

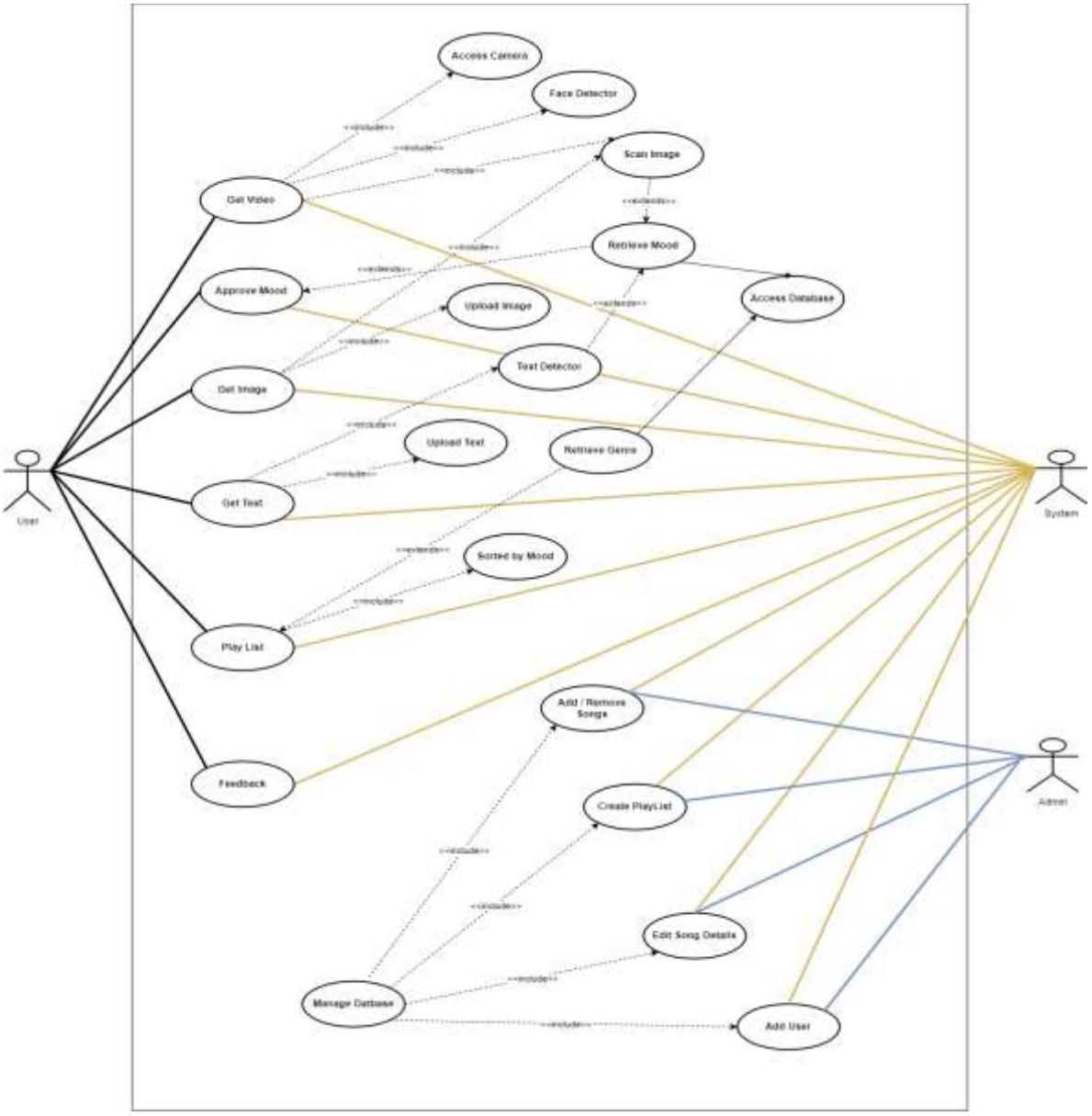

Figure 33 : Usecase Diagram

8. Database Design

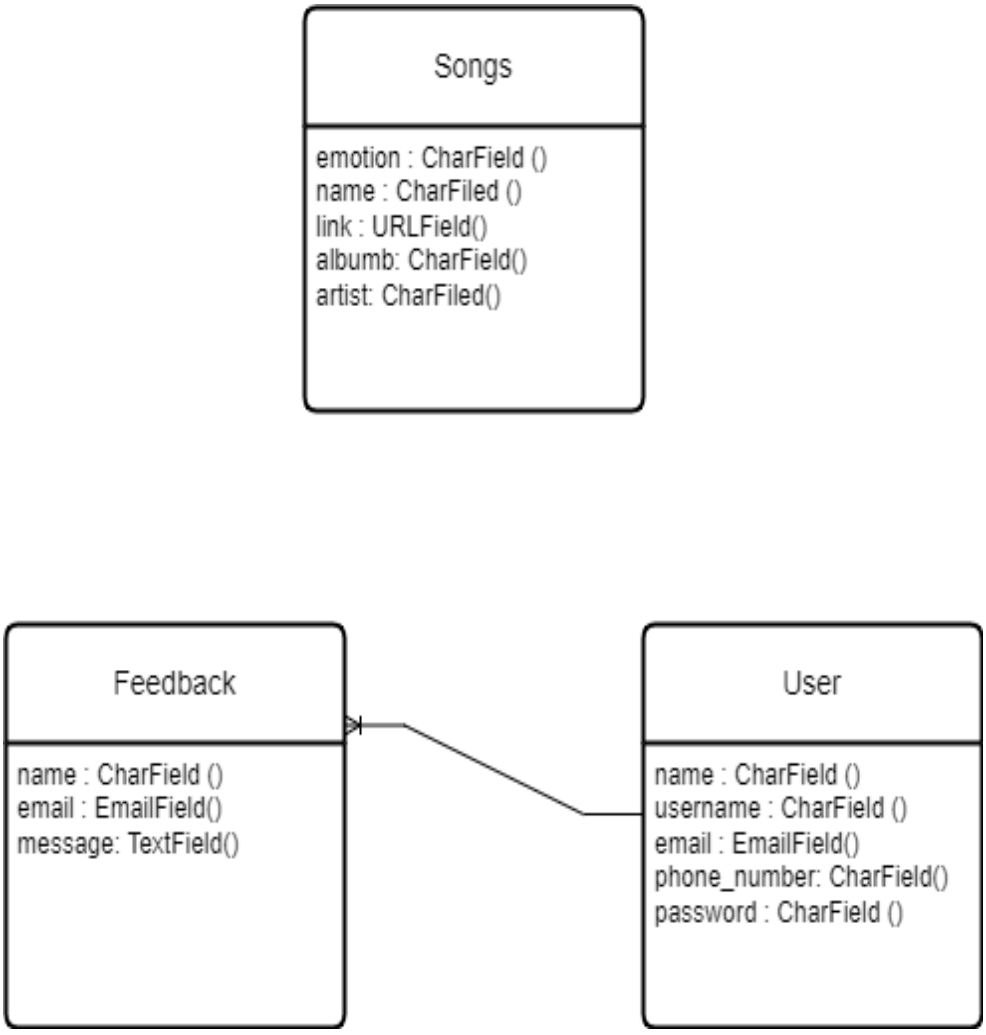

Figure 34 : Database Design

9. References

1. https://www.irjet.net/archives/V8/i7/IRJET-V8I7457.pdf
2. https://www.researchgate.net/publication/333816454_Emotion-Based_Music_Player_Emotion_Detection_from_Live_Camera
3. https://www.jetir.org/papers/JETIR2105803.pdf
4. https://ijcrt.org/papers/IJCRT2204529.pdf
5. https://sci-hub.se/10.1016/j.psychres.2013.03.001
6. https://sci-hub.se/10.1109/icpr48806.2021.9412591