RAJALAKSHMI ENGINEERING COLLEGE

**RAJALAKSHMI NAGAR, THANDALAM – 602 105**


RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

# CP23211 ADVANCED SOFTWARE ENGINEERING LAB

**Laboratory Record**
**Note Book**

Name: ……SARAVANAN S……. ……

Year / Branch / Section: …I / M.E -CSE…….. …….

University Register No. : ……2116230711009……

College Roll No. : ……230711009…….……

Semester: ……II…….……

Academic Year: ……2023 - 2024……

**RAJALAKSHMI ENGINEERING COLLEGE**

**RAJALAKSHMI NAGAR, THANDALAM - 602105**

**BONAFIDE CERTIFICATE**

Name: SARAVANAN S

Academic Year: 2023 - 2024 Semester: II Branch: M.E-CSE

Register No: 2116230711009

*Certified that this is the bonafide record of work done by the above student in the CP23211-Advanced Software Engineering Laboratory during the year 2023- 2024*

Signature of Faculty-in-charge

Submitted for the Practical Examination held on 22/06/2024

Internal Examiner External Examiner

# INDEX

# FACIAL EMOTION RECOGNITION FOR MUSIC GENERATION USING NLP

# OVERVIEW OF THE PROJECT:

Facial emotion recognition for music generation using NLP aims to develop an innovative system that interprets human emotions from facial expressions and generates corresponding music using Natural Language Processing (NLP). By integrating facial emotion recognition using Convolutional Neural Networks (CNNs) with NLP models like GPT-3 to translate detected emotions into descriptive texts or music instructions, the system can create emotionally resonant music. This real-time system, implemented with web technologies and user-friendly interfaces, has potential applications in therapy, entertainment, and personalized user experiences. The project focuses on enhancing the accuracy of emotion detection, the quality of generated music, and overall real-time performance to provide a seamless and engaging interaction for users.

# SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

**EXP.NO: 1** **DATE: 05.03.2024**

## CONTENTS

# FACIAL EMOTION RECOGNITON FOR MUSIC GENERATION USING NLP

## 1. Introduction

### 1.1 Purpose:

The purpose of the Facial Emotion Recognition for Music Generation using NLP project is to create a system that enhances human-computer interaction by interpreting human emotions from facial expressions and generating personalized music that reflects those emotions. This system aims to provide users with emotionally resonant music experiences tailored to their current emotional states, offering potential benefits in therapeutic settings, entertainment, and personal use. By accurately detecting emotions and using advanced NLP techniques to translate these emotions into music instructions, the project seeks to improve emotional expression, mood regulation, and overall user engagement through the power of music.

### 1.2 Scope:

The scope of the Facial Emotion Recognition for Music Generation using NLP project encompasses the development of a comprehensive system that integrates advanced facial emotion detection, Natural Language Processing (NLP), and music generation technologies. This project involves creating a robust model to classify emotions from facial expressions in real-time, translating these emotions into descriptive texts or music instructions using NLP, and generating corresponding music sequences that reflect the detected emotional states. The system aims to operate in real-time, providing instantaneous feedback through a user-friendly interface. Applications of this project include therapeutic settings to help individuals manage their emotions, enhancing user experiences in entertainment and gaming through emotionally responsive music, and creating personalized music playlists for personal enjoyment and mood regulation.

## 2. Overall Description

### 2.1 Product Perspective:

The Facial Emotion Recognition for Music Generation using NLP system can be developed as both a mobile application and a web-based application, accessible on both Android and iOS platforms.

## 2.2 Features

### 2.2.1. Real-Time Facial Emotion Detection

- **Camera Integration**: Uses the device's camera to capture facial expressions in real-time.
- **Emotion Classification**: Classifies emotions such as happiness, sadness, anger, surprise, and neutrality using a pre-trained Convolutional Neural Network (CNN).
- **Accuracy and Speed**: Provides quick and accurate detection to ensure seamless user experience.

### 2.2.2 Emotion-to-Text Conversion Using NLP

- **Descriptive Text Generation**: Converts detected emotions into descriptive texts that describe the emotional state.
- **Emotion Mapping**: Maps specific emotions to predefined text descriptions or music instructions.
- **NLP Models**: Utilizes advanced NLP models like GPT-3 for generating contextually relevant text.

### 2.2.3. Music Generation

- **Emotion-Based Music Composition**: Generates music sequences that match the detected emotional state.
- **Music Models**: Employs models like Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), or Generative Adversarial Networks (GANs) to produce high-quality music.
- **Dynamic Adjustment**: Adjusts tempo, melody, and harmony based on the detected emotions to ensure the music is emotionally resonant.

### 2.2.4. User Interface

- **Emotion Display**: Shows the detected emotion on the screen in real-time.
- **Music Player**: Provides controls for playing, pausing, and stopping the generated music.
- **Interactive Elements**: Includes buttons and sliders for user interaction and customization.

### 2.3 User Classes and Characteristics:

- General Users: Includes individuals who use the application for personal enjoyment and mood regulation.
- Therapists and Patients: Therapists who use the application as a tool in therapeutic settings and patients who benefit from music therapy.

## 3. Specific Requirements

### 3.1 Functional Requirements:

#### 3.1.1 Emotion Detection Module:

- **Facial Capture**: The system must be able to access the device's camera to capture real-time facial images.
- **Preprocessing**: The system must preprocess captured images (e.g., resizing, normalization) to ensure consistent input for emotion detection.
- **Emotion Classification**: The system must classify facial expressions into predefined emotional categories (e.g., happy, sad, angry, surprised, neutral) using a Convolutional Neural Network (CNN) or other suitable algorithms.
- **Real-Time Processing**: The system must detect and classify emotions in real-time with minimal latency.

#### 3.1.2 NLP Processing Module:

- **Emotion Mapping**: The system must map detected emotions to descriptive texts or music-related instructions.
- **Text Generation**: The system must use NLP models (e.g., GPT-3) to generate detailed text descriptions or instructions based on the detected emotions.
- **Contextual Relevance**: The generated text must be contextually relevant and accurately reflect the detected emotion.

#### 3.1.3 Music Generation Module:

- **Music Composition**: The system must generate music sequences that align with the emotion-specific instructions.

- **Model Utilization**: The system must use models like Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, or Generative Adversarial Networks (GANs) for music generation.
- **Emotion Alignment**: The generated music must be emotionally resonant and reflect the detected emotional state (e.g., sad music for sad emotions).

### 3.1.4 User Interface (UI):

- **Emotion Display**: The system must display the detected emotion to the user in real-time.
- **Music Player**: The system must provide controls for playing, pausing, stopping, and adjusting the volume of the generated music.
- **Interactive Elements**: The system must include buttons and sliders for user interaction and customization.
- **Feedback Collection**: The system must allow users to provide feedback on the accuracy of emotion detection and the quality of generated music.

## 3.2 Non-Functional Requirements:

### 3.2.1 Usability

- The user interface must be intuitive and easy to navigate for users of all technical proficiencies.
- The system must be accessible to users with disabilities, following WCAG (Web Content Accessibility Guidelines) standards.
- The application must provide immediate feedback to user actions to enhance the user experience.

### 3.2.2 Performance

- The system must have minimal latency, ideally under 200 milliseconds, between emotion detection and music generation to ensure a seamless real-time experience.
- The system must handle multiple user interactions simultaneously without performance degradation.
- The system must be scalable to support an increasing number of users and data without compromising performance.

### 3.2.3 Security

- The system must encrypt sensitive user data, including facial images and emotional data, both at rest and in transit.
- The system must implement robust authentication mechanisms to verify user identities and ensure proper authorization for accessing specific features.
- The system must comply with relevant data privacy regulations (e.g., GDPR, CCPA) to protect user information.

# 4. External Interface Requirements

## 4.1 User Interfaces:

**Emotion Detection Display**:

- The system must display the detected emotion in real-time on the user interface.
- The emotion label and a corresponding emoticon or graphic must be shown for clear understanding.

**Music Player Controls**:

- The system must provide standard music player controls, including play, pause, stop, and volume adjustment.
- Users must have the ability to skip to the next track or replay the current track.

**Interactive Elements**:

- The interface must include buttons and sliders for users to start/stop emotion detection and music generation.
- The system must provide customization options, such as selecting music genres or adjusting the intensity of the music.

**Feedback Mechanism**:

- The interface must allow users to rate the accuracy of emotion detection and the quality of the generated music.
- A text box or form should be provided for users to submit detailed feedback.

### 4.2 Hardware Interfaces:

**Camera**:

- The system must utilize the device's camera to capture facial expressions in real-time.
- It must support both front and rear cameras where applicable.

**Microphone**:

- If voice commands or additional audio cues are implemented, the system must use the device's microphone to capture these inputs.

### 4.3 Software Interfaces:

**Operating Systems**:

- The system must be compatible with major operating systems, including iOS, Android, Windows, and macOS.

**Web Browsers**:

- The web-based version of the system must support popular browsers like Chrome, Firefox, Safari, and Edge.

## 5. Conclusion

The external interface requirements ensure that the Facial Emotion Recognition for Music Generation using NLP system interacts effectively with users, hardware, software, APIs, data storage, and external services. These requirements focus on providing a user-friendly, secure, and seamlessly integrated experience, enabling the system to meet the diverse needs of its user base across various platforms and devices.

# SCRUM METHODOLOGY

**EXP.NO: 2** **DATE: 14.03.2024**

## 1. Introduction

The Facial Emotion Recognition for Music Generation Using NLP project involves multiple complex components, including emotion detection, natural language processing (NLP), and music generation. The Agile Scrum framework will manage this complexity by promoting adaptive planning, evolutionary development, early delivery, and continuous improvement.

## 2. Objectives

- The objective of implementing the Facial Emotion Recognition for Music Generation Using NLP project within the Scrum methodology framework is to deliver a high-quality, user-centered product incrementally and iteratively.
- This approach focuses on fostering collaboration, enhancing flexibility, and ensuring continuous improvement to meet user needs and business goals effectively.

## 3. Product Backlog Introduction

The product backlog is a dynamic list of features, enhancements, and fixes prioritized by the product owner. It serves as a roadmap for the development team.

## 4. Product Backlog

The Product Backlog is an ordered list of features, enhancements, and bug fixes that the product needs. In this project, it will include:

- User stories related to facial emotion detection (e.g., "As a user, I want the system to accurately detect my facial emotions in real-time").
- Tasks related to NLP processing and music generation.
- Technical tasks like integrating APIs, setting up the backend infrastructure, and ensuring data privacy and security.

## 5. User Stories

**As a User:**

- I want the system to accurately detect my facial emotions in real-time so that the music reflects my current mood.
- I want the system to recognize and differentiate between subtle changes in my facial expressions so that the music changes accordingly.
- I want to be notified if the system fails to detect my facial emotions so that I can adjust my position or lighting conditions.

## 6. Sprint

- A time-boxed iteration during which a set of user stories is implemented and tested.

## 7. Sprint Planning

During Sprint Planning, the team discusses and selects Product Backlog items to work on during the Sprint. They break down these items into tasks and estimate the effort required. The goals are to:

- Identify tasks related to improving the emotion detection algorithm's accuracy.
- Plan integration of the NLP model for generating music-related text instructions.

## 8. Sprint Backlog

The Sprint Backlog contains items selected from the Product Backlog to be completed in the current Sprint. It includes:

- Detailed tasks to be accomplished during the Sprint.
- Clear definitions of done for each task to ensure they meet the quality standards.

## 9. Sprint Retrospective

In the Sprint Retrospective, the team reflects on the Sprint process and identifies areas for improvement. The goal is to:

- Continuously improve team dynamics, processes, and tools.
- Discuss what went well, what could be improved, and actionable steps for future Sprint

## 10. Sprint Review

During the Sprint Review, the team presents the completed work to stakeholders and Gathers feedback. This helps:

- Demonstrate progress, such as a functional demo of the emotion detection module.
- Collect valuable feedback for refining the next Sprint's tasks.

## 11. Software Used

- Development Platform: PyCharm, VS Code, Xcode, OpenCV, Dlib, spaCy, User Interface.

## 12. Conclusion

The development platform for Facial Emotion Recognition for Music Generation using NLP encompasses a diverse set of tools and frameworks tailored to each aspect of the project. From development environments and version control systems to specific libraries for emotion detection, NLP, and music generation, each component is carefully chosen to ensure robust, scalable, and efficient development.

# USER STORIES

**EXP.NO: 3** **DATE: 26.03.2024**

## As a User:

1. **Emotion Detection**

**User Story 1:** I want the system to accurately detect my facial emotions in real-time so that the music reflects my current mood.

**Acceptance Criteria:**

The system must accurately identify emotions such as happiness, sadness, anger, surprise, and neutrality within 2 seconds of capturing the image.

**User Story 2:** I want the system to recognize and differentiate between subtle changes in my facial expressions so that the music changes accordingly.

**Acceptance Criteria:**

The system must detect and reflect changes in emotions with an accuracy of at least 85% and adjust the music within 5 seconds.

2. **User Interface**

**User Story 1:** I want a clear and intuitive interface to start and stop emotion detection easily so that I can control when the system interacts with me.

**Acceptance Criteria:**

The interface must have clearly labeled buttons for starting and stopping emotion detection, and these buttons must respond within 1 second.

**User Story2:** I want to see a visual representation of my detected emotion on the screen so that I understand what the system perceives.

**Acceptance Criteria:**

The interface must display the detected emotion as a label and an emoticon within 1 second of detection.

**3. Music Generation**

**User Story1:** I want the system to generate music that reflects my current emotional state so that the music enhances my mood.

**Acceptance Criteria:**

The system must generate music that aligns with the detected emotion within 5 seconds of detection.

**4. Feedback and Improvement**

**User Story 1:** I want to provide feedback on the accuracy of emotion detection so that the system can improve over time.

**Acceptance Criteria:**

The interface must have an option for users to rate the accuracy of detected emotions and submit feedback, with submissions processed immediately.

**5. Data Privacy and Security**

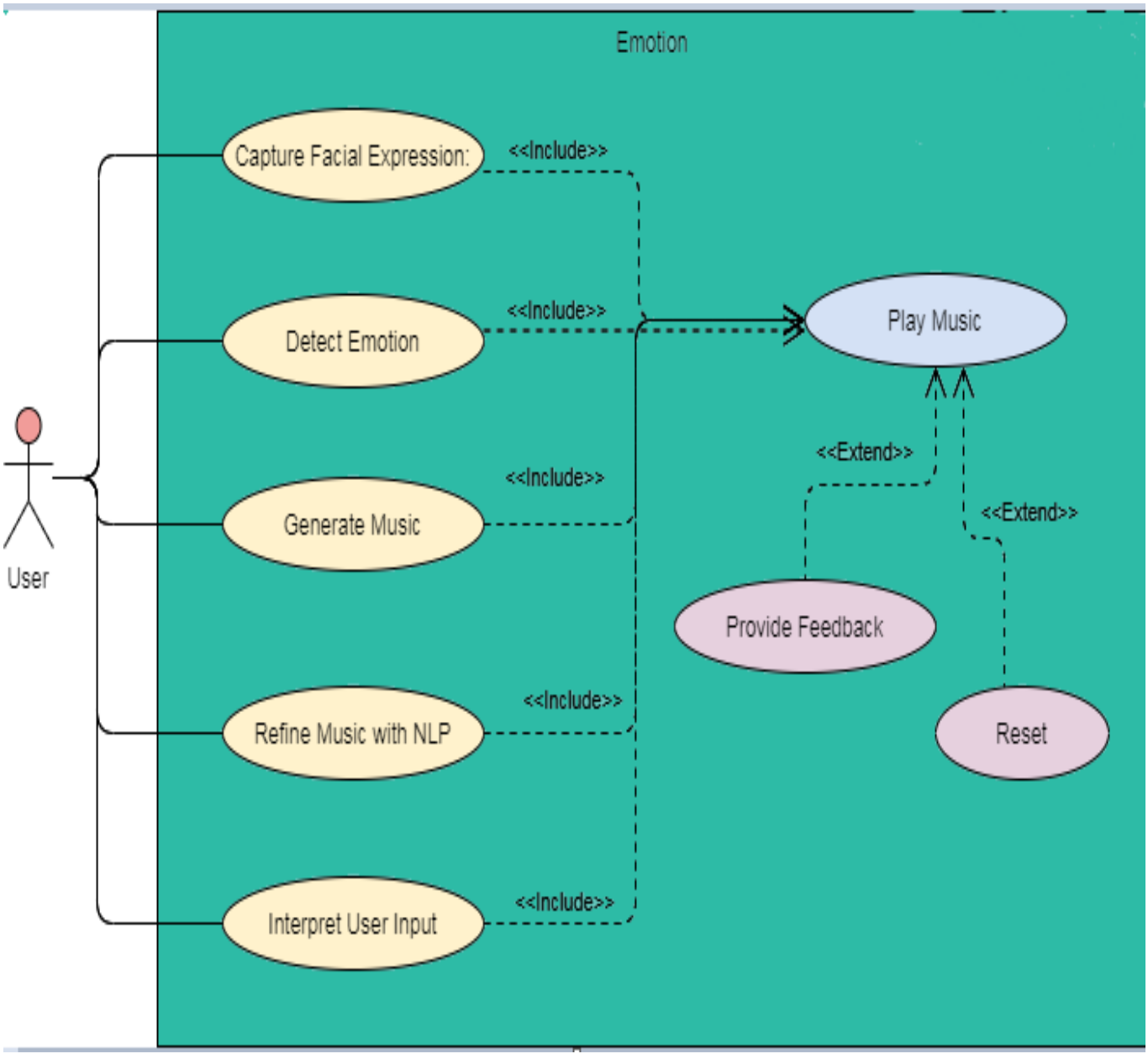
**User Story 1:** I want to know how my data is being used and stored so that I feel secure using the system.

**Acceptance Criteria:**

The system must provide clear information on data usage and storage practices and obtain user consent before collecting data.

# USE CASE DIAGRAM

**EXP.NO: 4** **DATE: 04.04.2024**



# NON-FUNCTIONAL REQUIREMENTS

**EXP.NO: 5** **DATE: 16.04.2024**

**1. Performance**

- Latency: The system should have low latency between emotion detection and generation music to provide a real-time experience for the user.
- Accuracy: The facial recognition and emotion analysis components should achieve a high degree of accuracy to ensure the generated music accurately reflects the user's emotional state.

**2. Security**

- Privacy: The system must ensure the user's facial data is collected, stored, and processed securely following data privacy regulations.
- Authentication: The system may implement user authentication mechanisms to restrict unauthorized access.

**3. Usability**

- Ease of Use: The user interface should be intuitive and easy to navigate for users with varying technical backgrounds.
- Customizability: The system may allow users to customize certain aspects of the music generation process, such as preferred genres or instruments.

**4. Reliability**

- Availability: The system should be highly available to process user requests consistently with minimal downtime.
- Fault Tolerance: The system should be designed to handle errors gracefully and recover from failures without significant data loss.

**5. Scalability**

The system should be scalable to accommodate an increasing number of users and data without performance degradation.

**6. Maintainability**

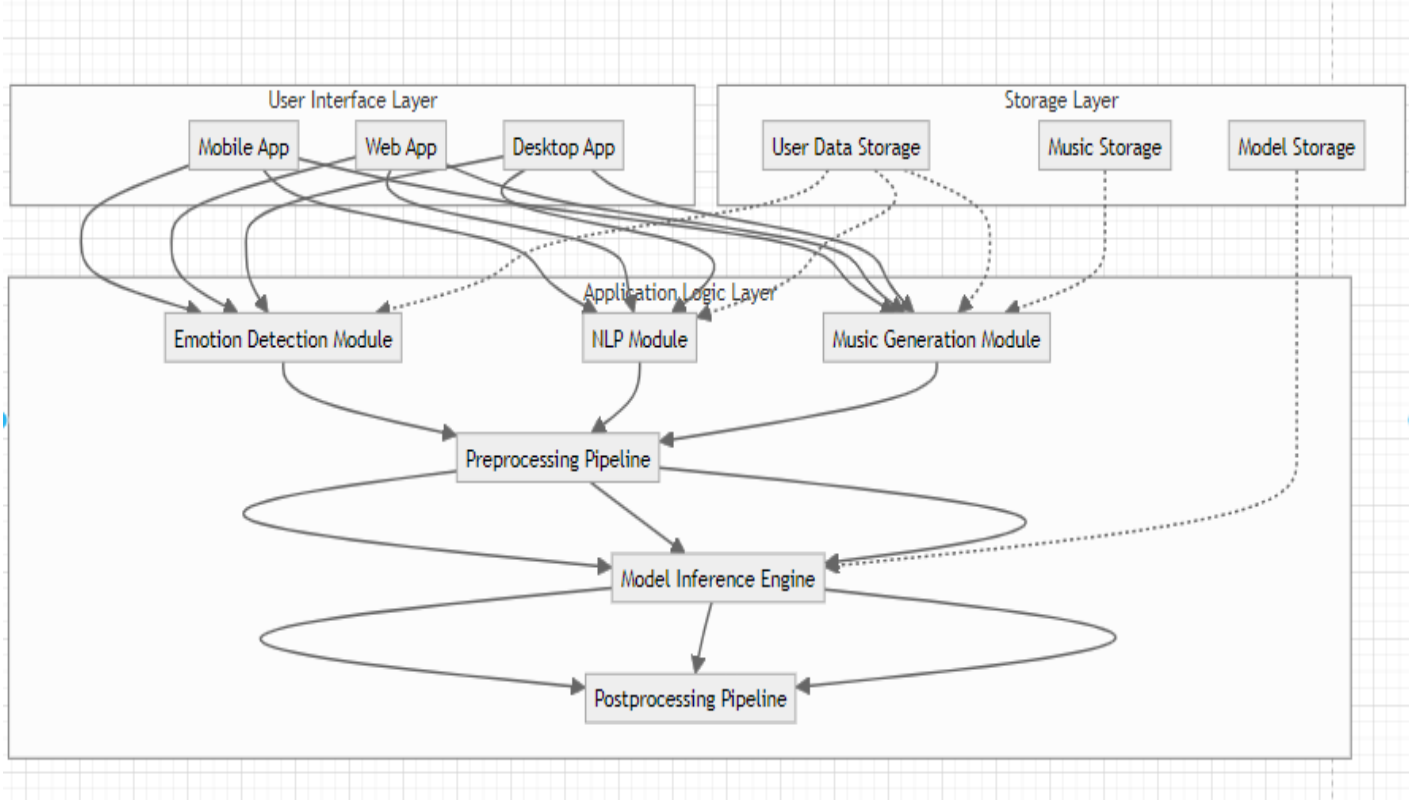
- Modular Design: The system should be designed with modular components to facilitate easier maintenance, updates, and future enhancements.
- Documentation: Comprehensive documentation should be provided for developers and system administrators to understand the system's functionality and troubleshoot issues.

**7. Compliance**

- Data Privacy Regulations: The system must comply with data privacy regulations such as GDPR, CCPA, and others applicable in different regions.
- Industry Standards: The system should adhere to relevant industry standards for security, accessibility, and interoperability.
- Auditability: The system should maintain detailed logs and records of user interactions, system processes, and access controls to support audits and compliance checks.
- Legal Requirements: The system must comply with all relevant legal requirements regarding data collection, storage, processing, and user consent.

# OVERALL PROJECT ARCHITECTURE

**EXP.NO: 6** **DATE: 25.04.2024**




## Layers:

The overall project architecture for the Facial Emotion Recognition for Music Generation using NLP project. This architecture consists of three main layers: User Interface Layer, Application Logic Layer, and Storage Layer.

## Components:

### 1. User Interface Layer:

- **Mobile App:** A mobile application for users to capture images, listen to generated music, and provide feedback.
- **Web App:** A web-based application offering similar functionality as the mobile app.
- **Desktop App:** A desktop application providing the same core functionalities.

### 2. Application Logic Layer:

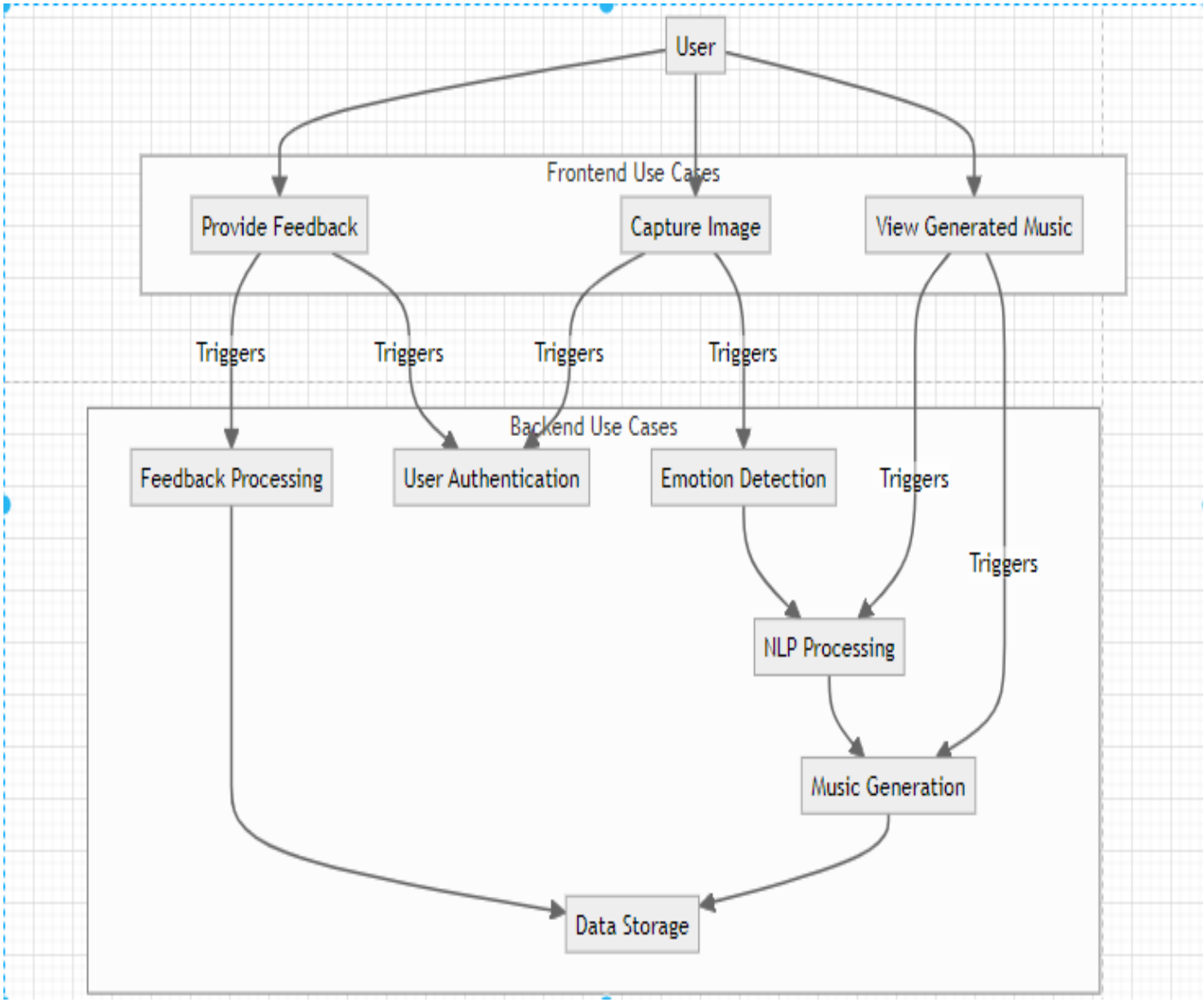
- **Emotion Detection Module:** Handles the detection of emotions from captured facial images.
- **NLP Module:** Processes detected emotions to generate music-related parameters.
- Music Generation Module: Generates music based on the parameters from the NLP module.
- **Preprocessing Pipeline:** Prepares the input data (e.g., facial images) for processing.
- Model Inference Engine: Runs the machine learning models to infer emotions and process NLP tasks.
- **Postprocessing Pipeline:** Finalizes the output data (e.g., generated music) before it is stored or presented to the user.

### 3. Storage Layer:

- **User Data Storage:** Stores user information, preferences, and feedback.
- **Model Storage:** Stores machine learning models used for emotion detection and NLP.
- **Music Storage:** Stores generated music files and related metadata.

# BUSINESS ARCHITECTURE DIAGRAM

**EXP.NO: 7** **DATE: 02.05.2024**



## Actor:

User: Represents the primary user who interacts with the application.This actor is involved in all frontend use cases.

## Frontend Use Cases:

- **Capture Image:** Allows the user to capture an image for emotion detection.
- **View Generated Music:** Enables the user to view and listen to the generated music based on detected emotions.
- **Provide Feedback:** Users can provide feedback on the generated music.

## Backend Use Cases:

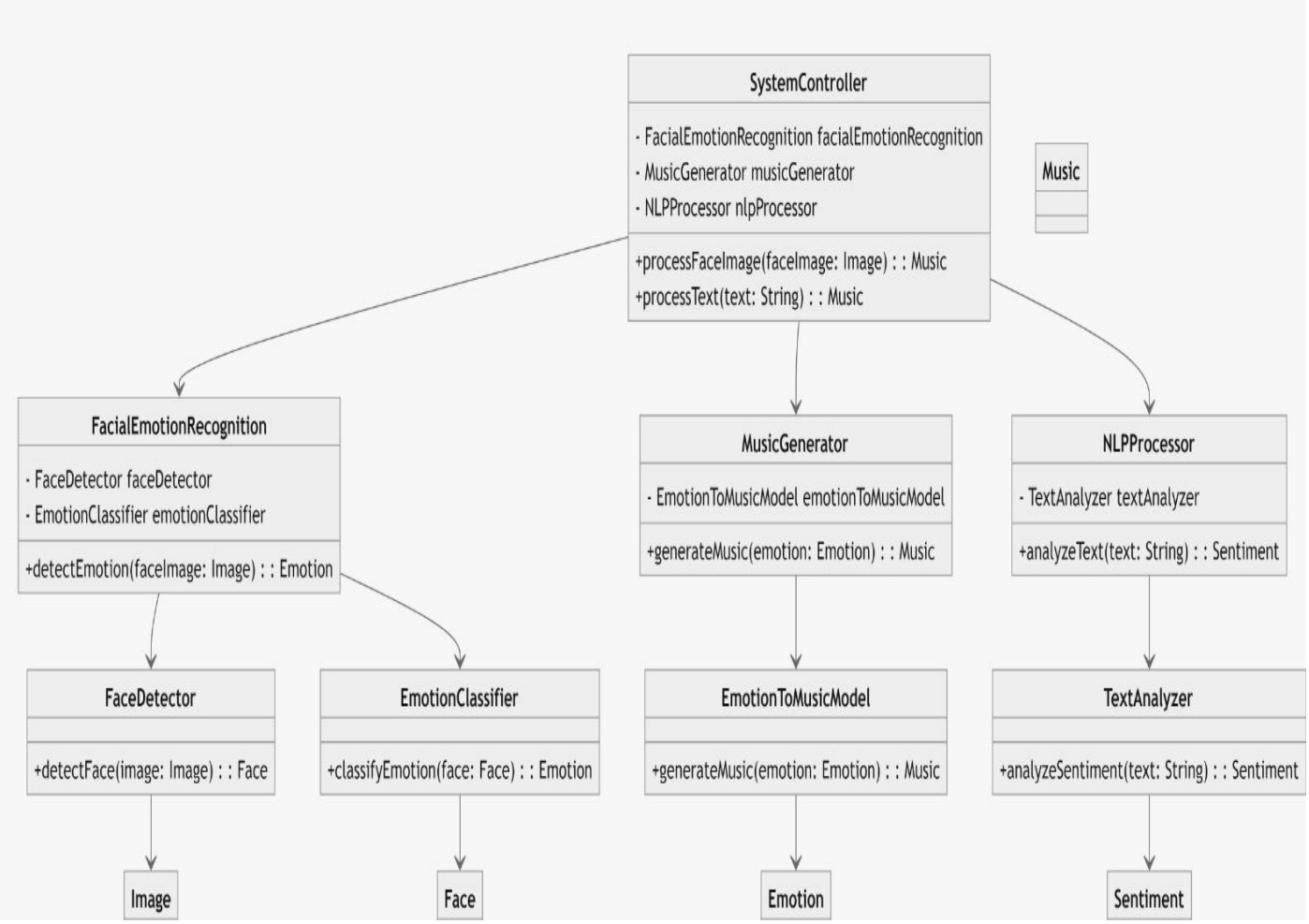
- **Emotion Detection**: Detects emotions from the captured image.
- **NLP Processing:** Processes the detected emotions to determine music generation parameters.
- **Music Generation:** Generates music based on the parameters obtained from NLP processing.
- **User Authentication:** Manages user authentication to ensure secure access.
- **Data Storage:** Stores user data, generated music, and feedback.
- **Feedback Processing:** Processes feedback provided by users to improve the system.

### Triggers

The arrows indicate the triggers between frontend and backend use cases. For instance, capturing an image triggers emotion detection, and viewing generated music involves NLP processing and music generation.

# CLASS DIAGRAM

**EXP.NO: 8** **DATE: 07.05.2024**



**Classes:**

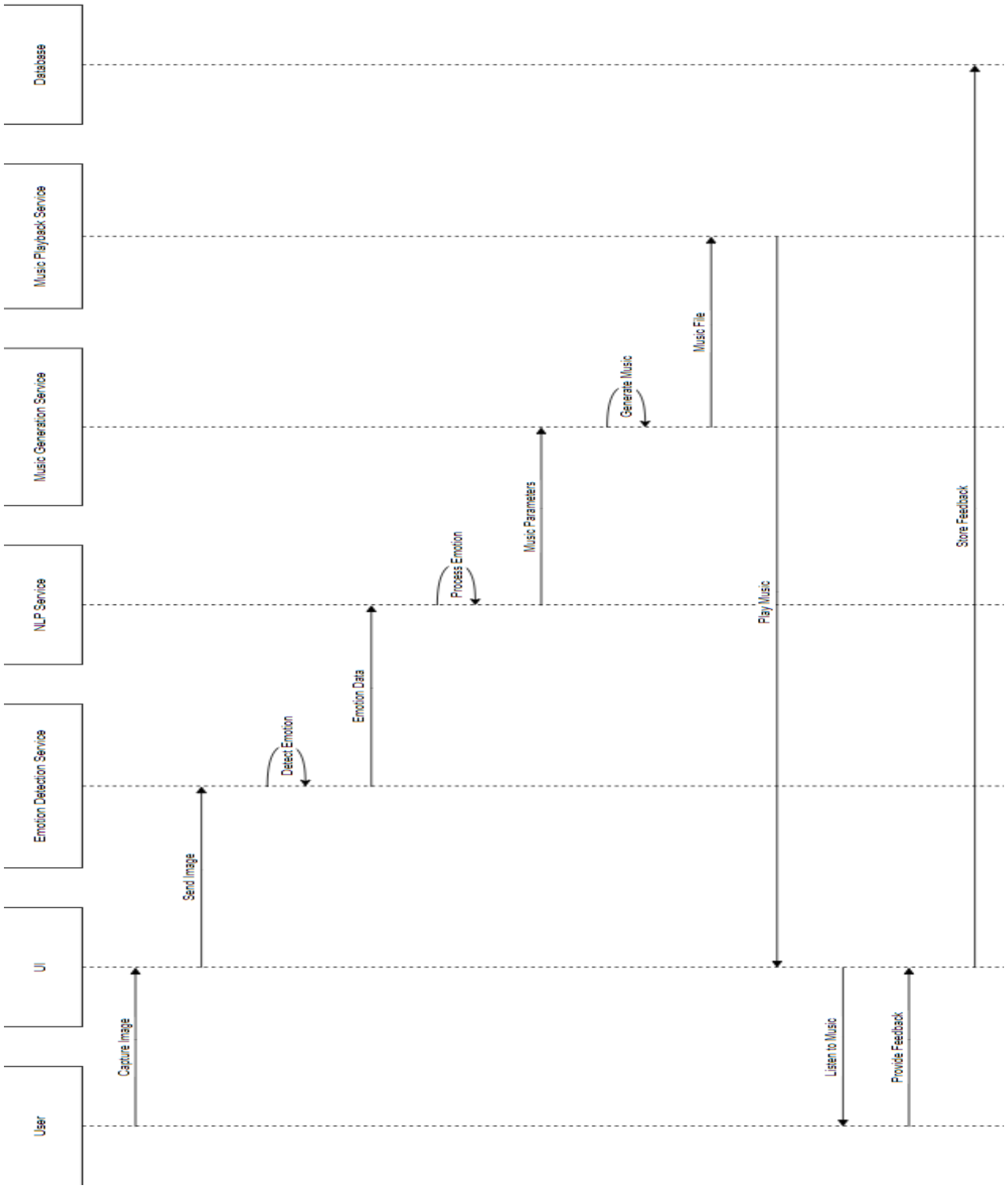
- **SystemController:** processFaceImage(faceImage: Image): Music: Processes a face image to generate music based on detected emotion.
- **processText(text: String):** Music: Processes text to generate music based on detected sentiment.
- **Facial Emotion Recognition:** This component detects and classifies emotions based on facial expressions.
- **Music Generation:** This component generates music that aligns with the detected emotion.
- **NLP Processing:** This component may involve processing textual data to enhance emotion detection or music generation.
- **Music Preferences:** Stores the music preferences derived from the detected emotions.
- **Music Selector:** Selects appropriate music based on the generated preferences.
- **Music Player:** Plays the selected music for the user.

**Relationships:**

- Facial Emotion Recognizer has an association with Emotion (returns detected emotions).
- NLP Processor has an association with Music Generator (uses it to generate music).
- Music Generator uses text data from NLP Processor to generate music.

# SEQUENCE DIAGRAM

**EXP.NO: 9** **DATE: 16.05.2024**



## Actor:

**User:** The user interacting with the application.

## System Components:

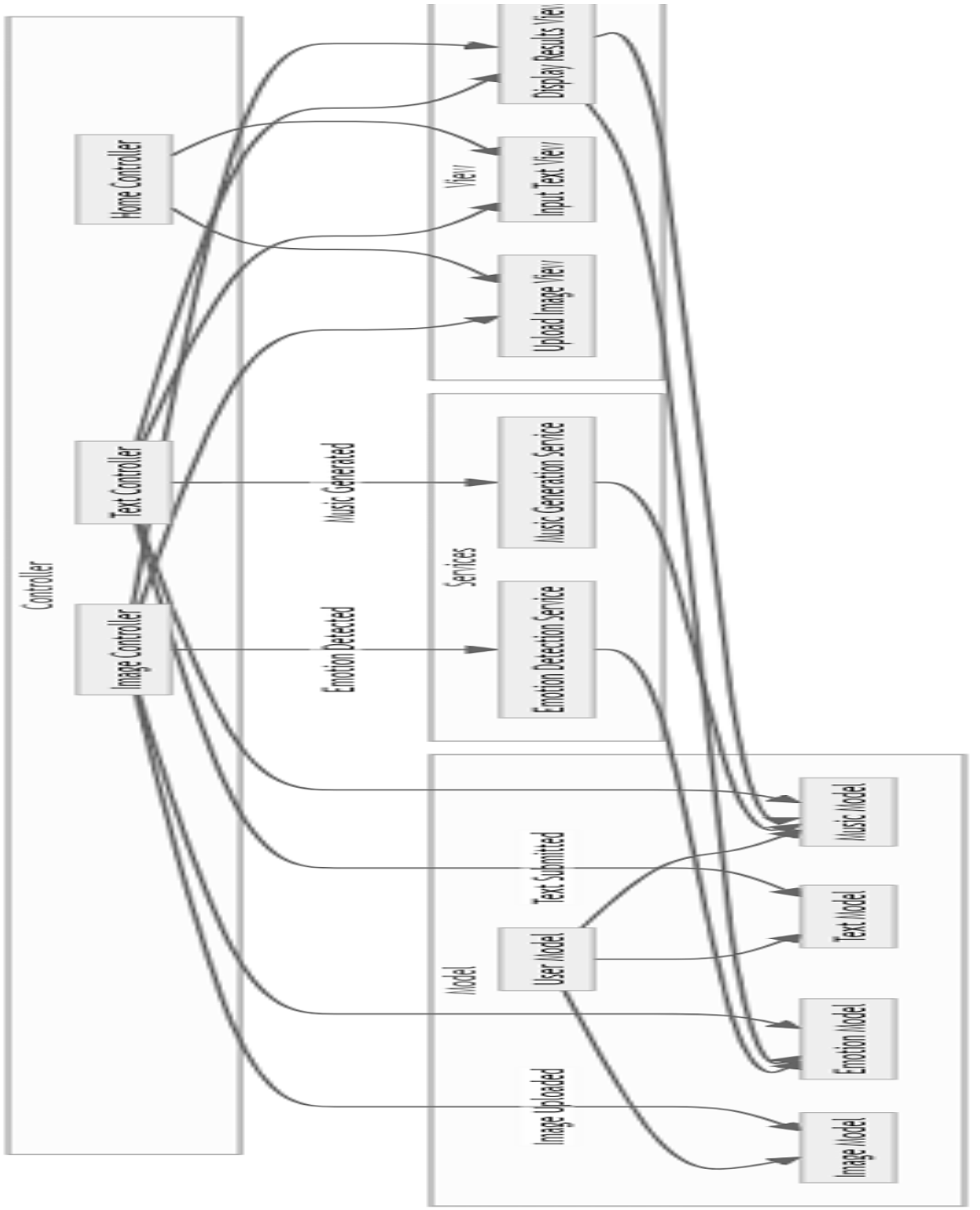
- **User:** The person using the application.
- **User Interface (UI):** The frontend interface, including mobile, web, or desktop application.
- **Emotion Detection Service:** Backend service that processes the image to detect facial emotions.
- **NLP Service:** Backend service that processes detected emotions to generate music-related text or parameters.
- **Music Generation Service:** Backend service that creates music based on the parameters from the NLP service.
- **Music Playback Service:** Component that handles playing the generated music.
- **Database:** Stores user data, preferences, feedback, and generated music.

## Sequence of Events:

- **Capture Image**: The User captures a facial image using the UI (mobile, web, or desktop application).
- **Send Image:** The UI sends the captured image to the Emotion Detection Service.
- **Detect Emotion:** The Emotion Detection Service processes the image to detect facial emotions. It sends the detected emotion data to the NLP Service.
- **Process Emotion:** The NLP Service processes the detected emotion to generate music-related text or parameters. These parameters are sent to the Music Generation Service.
- **Generate Music:** The Music Generation Service uses the parameters to generate a music file. The generated music file is sent to the Music Playback Service.
- **Play Music:** The Music Playback Service plays the generated music to the User through the UI.
- **Provide Feedback:** The User listens to the music and provides feedback through the UI. The feedback is sent to the Database for storage and future improvement of the system.

# ARCHITECTURAL PATTERN (MVC)

**EXP.NO: 10** **DATE: 28.05.2024**



## Model:

- **User Model:** Represents the user data.
- **Image Model:** Represents the image data.
- **Emotion Model:** Represents the detected emotion data.
- **Text Model:** Represents the text input data.
- **Music Model:** Represents the generated music data.

## View:

- **Upload Image View:** The user interface for uploading images.
- **Input Text View:** The user interface for inputting text.
- **Display Results View:** The user interface for displaying results such as detected emotions and generated music.

## Controller:

- **Home Controller:** Manages the home page and navigation to other views.
- **Image Controller:** Handles image upload, invokes emotion detection, and updates the view.
- **Text Controller:** Handles text input, invokes music generation, and updates the view.

## Relationships:

The Model, View, and Controller components interact as follows:

- Home Controller directs users to the Upload Image View and Input Text View.
- Image Controller manages the Upload Image View, processes uploaded images, and interacts with Image Model, Emotion Detection Service, and Emotion Model. It updates the Display Results View.
- Text Controller manages the Input Text View, processes text input, and interacts with Text Model, Music Generation Service, and Music Model. It updates the Display Results View.
- The Display Results View accesses data from Emotion Model and Music Model to present results to the user.
- Emotion Detection Service and Music Generation Service handle the core processing logic and update the respective models.