Detailed Design Document: Soundtrack Fitting for Dramatic Clips

---

1. Introduction

Purpose

This document provides a detailed design for the **"Adapting Music to Dramatic Short Videos"** project. It outlines the system architecture, components, data models, and algorithms to guide the development process and ensure the successful implementation of the system.

---

2. System Overview

- The system leverages advanced AI/ML techniques to analyse dramatic scenes and suggest matching music tracks. Key components include:
- **Scene Analysis:** Detects visual and auditory cues such as brightness, motion intensity, and dialogue tone.
- **Music Matching:** Matches scenes with tracks from a curated music library.
- **Customization:** Allows users to fine-tune suggestions.
- **Exporting:** Outputs the final video with the adapted music.

---

3. Design constraints

Assumptions

- Users will provide videos that are up to 3 minutes.
- Users will upload dramatic video clips only, not other genres.
- The music library will include tracks suited specifically for dramatic themes (e.g., suspense, sadness, triumph).
- Existing APIs and libraries for video and audio analysis will be sufficient for the project requirements.

Constraints

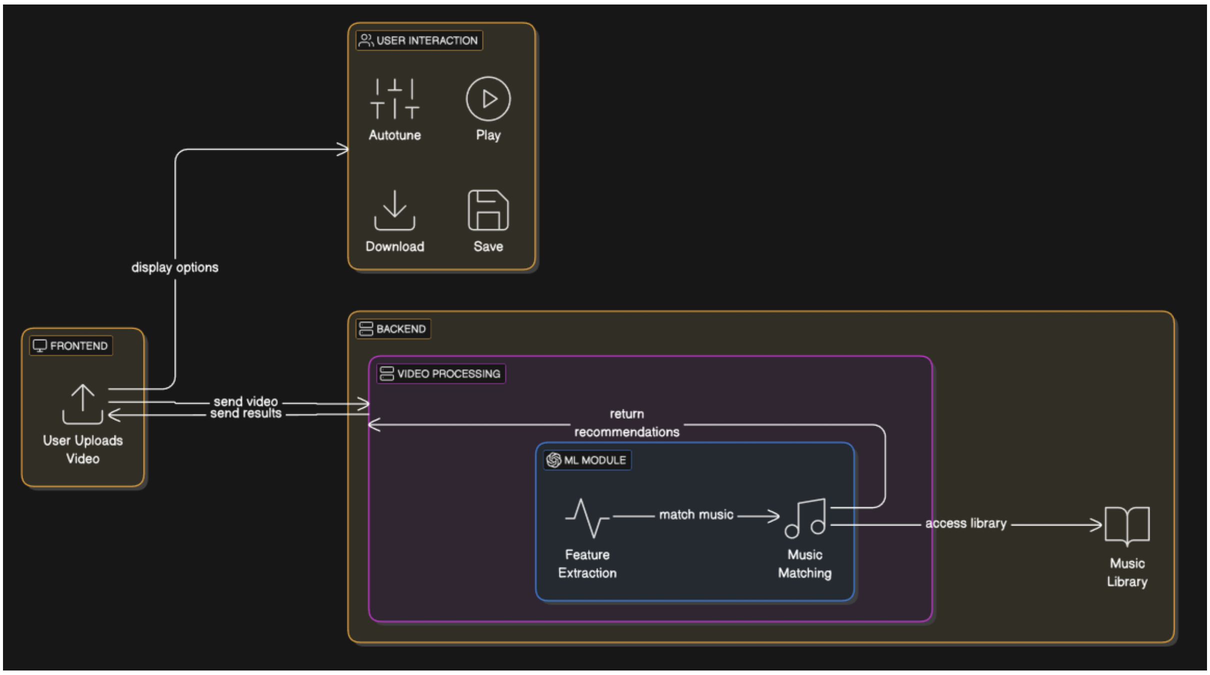
- The system will only analyze and match music for **dramatic clips** to ensure focus and feasibility within the time constraints.
- The music library will be limited to royalty-free tracks to avoid legal issues.
- Support a minimum of 5 concurrent users.
- The system must process video uploads and suggest appropriate music tracks in reasonable time per clip.

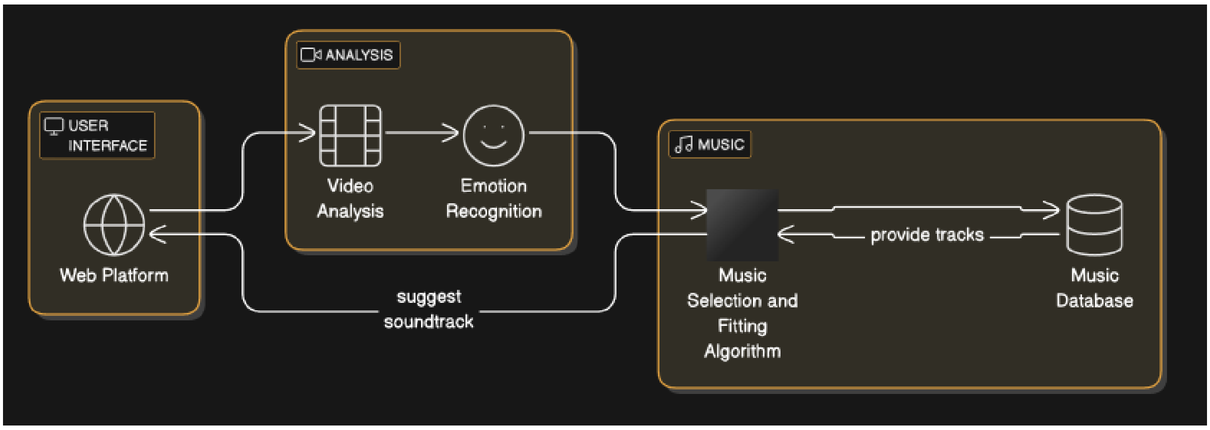
Standards

- Training data for machine learning models will be preprocessed to meet consistency and quality requirements.
- The user interface will be intuitive and user-friendly, adhering to basic UI/UX design principles.
- Follow language-specific guidelines such as PEP 8 for Python, or ESLint for JavaScript, ensuring readable and maintainable code.

---

4. System Architecture





User Interface:

- Web platform for user interaction
- Displays suggested soundtracks
- Allows user input/control

Video Analysis & Emotion Recognition:

- System analyzes input video content
- Detects emotional elements and mood

Music Processing:

- Music database stores tracks
- Selection algorithm matches music to detected emotions
- Suggests appropriate soundtrack based on analysis

---

5. Component Design

Scene Analysis Module

- **Input:** Video file.
- **Process:** Extracts frames and audio; analyses features using OpenCV (visual) and LibROSA (audio).
- **Output:** Dramatic parameters (e.g., suspense, romance, action).

Music Matching Module

- **Input:** Scene analysis results.
- **Process:** Matches scene parameters with tracks in the music library using content-based and collaborative filtering.
- **Output:** Three music recommendations.

Customization Module

- **Features:** Adjust tempo, volume, and fade-in/out effects.
- **Tools:** FFMPEG for audio processing.

  (database explanation)

---

6. Database Design

VideoClip:

- Central entity managing video files
- Contains functions for analyzing visual/audio features
- Handles music fine-tuning

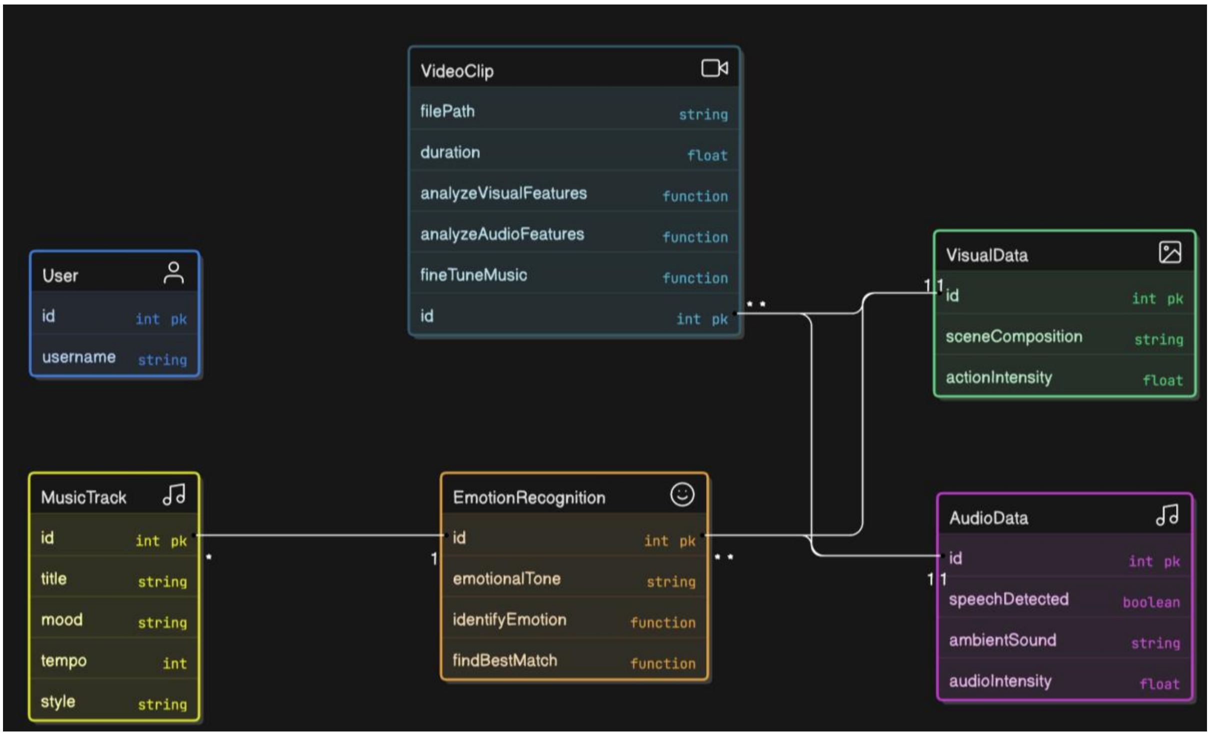
Related Data Entities:

- VisualData: Tracks scene composition and action intensity

- AudioData: Monitors speech, ambient sound, and audio intensity
- EmotionRecognition: Processes emotional tone and matches content
- MusicTrack: Stores music metadata (title, mood, tempo, style)
- User: Basic user identification and management

The relationships show:

- One VideoClip can have multiple Visual/Audio data points
- EmotionRecognition links VideoClip analysis with MusicTrack selection
- Each entity has a primary key (pk) for unique identification



7. Algorithms and Processing

Scene Analysis Algorithm

1. Extract frames and audio.
2. Analyse visual cues (brightness, motion intensity).
3. Analyse auditory cues (tempo, pitch).
4. Classify dramatic theme (e.g., suspense, romance).

Music Recommendation Algorithm

1. Match scene parameters with music metadata.
2. Apply collaborative filtering based on user preferences.
3. Rank and suggest tracks.

Customization Algorithm

1. Apply user adjustments (e.g., volume, tempo).
2. Re-process audio using FFMPEG.

---

8. User Interface Design

Features

- **Scene Upload:** Drag-and-drop functionality.
- **Analysis Results:** Display dramatic parameters and recommendations.
- **Customization Tools:** Sliders for volume and tempo adjustments.
- **Preview and Export:** Preview the adapted video and download.

Design Considerations

- Responsive layout for all devices.
- Minimalistic and intuitive design.

Algorithm Description

1. Video Analysis Algorithm

- **Algorithm:** The video analysis algorithm extracts features from both the visual and audio components of the dramatic clip. Visual features such as scene composition, facial expressions, object movements, and camera angles are detected using Convolutional Neural Networks (CNNs). Audio features, such as speech, ambient sound, and intensity variations, are extracted using Mel-frequency cepstral coefficients (MFCCs). The combination of these features provides insights into the emotional tone and intensity of the video clip.
- **Time Complexity:** Depends on the resolution of the video and the length of the clip. For a single frame, the complexity of CNN-based visual analysis is O(n * m), where n is the number of pixels in the frame, and m is the number of frames processed. Audio feature extraction may require O(p * q) time, where p is the number of audio samples, and q is the number of extracted features.

2. Emotion Recognition Algorithm:

- **Algorithm:** After extracting the features from the video, an emotion recognition model uses machine learning technique Recurrent Neural Networks (RNNs) to classify the emotional tone of the video clip (e.g., tension, sadness, excitement). This model uses both the visual and audio features as input to determine the emotional state of the scene.
- **Time Complexity:** The time complexity for emotion recognition is determined by the complexity of the chosen model and the size of the input features. For an SVM, the time complexity is O(n^2 * m) to O(n^3 * m), depending on the number of features (n) and the

size of the dataset (m). For neural networks like RNNs, the time complexity is generally O(n * m * t), where t is the number of time steps in the input sequence.

3. Music Database and Matching Algorithm:

- **Algorithm:** The music database consists of a large collection of tracks categorized by mood, tempo, genre, and emotional cues. Each track is tagged with metadata based on these categories. When a video clip is analyzed, the system uses similarity metrics such as cosine similarity, Euclidean distance, or dynamic time warping (DTW) to match the emotional features of the video with the closest fitting music tracks in the database. A machine learning model, like a k-Nearest Neighbors (k-NN) classifier, can be used to identify the most similar tracks.
- **Time Complexity:** The time complexity of matching involves comparing the emotional feature vectors of the video with the music database. If there are 'm' music tracks and 'n' features, the matching time complexity is O(m * n) for each video clip. The use of optimized data structures such as KD-trees or Ball-trees can reduce the matching time to O(log(m) * n) in the case of high-dimensional data.

4. Music Selection Algorithm:

- **Algorithm:** Based on the matched music track(s) from the database, the selection algorithm evaluates several factors, such as tempo, harmony, and intensity, to ensure the selected music fits the video clip's flow and pacing. The music selection is adjusted to complement changes in emotional intensity throughout the clip. This step ensures a smooth transition and alignment between the video and audio.
- **Time Complexity:** The complexity of the music selection step is typically O(n * m), where 'n' is the number of possible music tracks, and 'm' is the number of evaluated attributes (tempo, harmony, etc.). Optimization techniques can reduce the complexity depending on the number of potential tracks and attributes.

5. Fine-Tuning Suggestions Algorithm (User Customization):

- **Algorithm:** After the initial music suggestions are provided, users can fine-tune the selection based on personal preferences or specific adjustments to the video’s emotional tone. The fine-tuning process allows users to adjust the music's tempo, intensity, mood, or style.
- **Time Complexity:** The time complexity for fine-tuning depends on the extent of the adjustments made by the user.

6. User Interface (UI) Algorithm:

- **Algorithm**: The user interface algorithm allows users to upload video clips, view the corresponding soundtrack suggestions, and make adjustments to fine-tune the music. The system processes the uploaded clip, runs the video analysis, emotion recognition, and music matching steps, and displays the results to the user in a web-based application. Users can interact with sliders or dropdowns to adjust the music's tempo, mood, and intensity.

- **Time Complexity:** The time complexity of the UI algorithm mainly depends on the backend processing of the video and music matching steps. The complexity is generally dominated by the video analysis and emotion recognition algorithms, resulting in a time complexity of O(m * n) for each video uploaded, where 'm' is the length of the video and 'n' is the number of features or tracks.