HIGH LEVEL DESIGN DOCUMENT

MUSIC RECOMMENDER SYSTEM WITH SENTIMENT ANALYSIS

UE20CS390A – Capstone Project Phase – 1

Submitted by:

Reshmi Pradeep	**PES2UG20CS270**
Alekhya Sundari R Nanduri	**PES2UG20CS900**
D. Mrudula	**PES2UG20CS905**
Harshitha Golla	**PES2UG20CS910**

Under the guidance of

Dr. L. Kamatchi Priya
Professor
PES University

January - May 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

TABLE OF CONTENTS

1. Introduction

This document provides a high level overview of the design for a music recommendation system with sentiment analysis. The system includes a web scraper to gather data from social media platforms, and a pre-processing step to clean and transform the data for classification of emotions which is then used in the recommendation algorithms. This project utilizes various machine learning techniques such as Content-Based Filtering with TF-IDF, GLoVe, USE and DCNN to recommend music based on the emotion, providing a personalized and engaging user experience. The system utilizes both collaborative and content-based filtering techniques to provide personalized recommendations to users based on their listening history, ratings, and emotions associated with the music they listen to.

The design incorporates a modular architecture with separate components for each step in the recommendation process. The system is designed to be scalable and able to handle large volumes of data efficiently. This document outlines the design and implementation details for each component, along with the data flow and relationships between components.

This document is intended for developers and stakeholders involved in the development of the music recommendation system, to provide an overview of the system's design and functionality.

2. Current System

All the existing music recommender systems don't take into account a user's emotions or personality traits. They are mainly based on their history, similarities with users etc. Some of the most popular music recommendation systems today are based on collaborative filtering, content-based filtering, and hybrid approaches that combine elements of both.

Collaborative filtering is a technique that involves analyzing user behavior, such as the songs they listen to, and identifying patterns in that behavior to make recommendations. This method is based on the idea that users who have similar listening habits are likely to enjoy similar music. Collaborative filtering systems typically recommend songs or artists that other users with similar tastes have enjoyed.

Content-based filtering, on the other hand, focuses on the characteristics of the music itself, rather than on user behavior. This method involves analyzing features such as genre, tempo, and mood to recommend music that is similar to what the user has previously enjoyed. Content-based filtering systems may also take into account other factors, such as lyrics, artist biographies, and album reviews, to make recommendations.

Hybrid approach combines elements of both collaborative and content-based filtering to provide more accurate and diverse recommendations. These systems analyze both user behavior and music characteristics to create a more complete picture of the user's preferences and make more personalized recommendations.

3. Design Considerations

3.1. Design Goals

The primary goal of the music recommendation system is to help users discover new music based on their sentiment expressed in tweets. The system should provide a personalized and engaging experience that helps users find music that matches their mood and preferences. The system should be user-friendly, easily accessible, and prioritize user privacy and data security.

The newly proposed system provides a personalized music recommendation system based on user sentiment analysis and feedback, which improves the accuracy and relevance of music recommendations and makes it more personalized and helps users to discover new music that aligns with their preferences and moods. The system uses sentiment analysis and machine learning algorithms to provide accurate and personalized recommendations.

Quality of service characteristics:

- Availability: The music recommendation system should be designed in such a way where the downtime is minimized.
- Security and privacy: The system should prioritize user privacy and ensure that user data is stored securely and handled responsibly.
- Speed: The system should provide recommendations quickly and efficiently, with minimal lag or delay.

The design of the music recommendation system with sentiment analysis should prioritize the following guidelines:

- Personalization: The system should provide personalized recommendations based on the user's mood and preferences.
- User-friendliness: The system should be easy to use, with a simple and intuitive interface.
- Security: The system should ensure that user data is stored securely and handled responsibly.
- Maintainability and scalability: The system should be designed for easy maintenance and scalability, with robust architecture and efficient data management.

The design of the music recommendation system with sentiment analysis is guided by the following principles:

- Emotional Classification: The system uses machine learning algorithms to identify the user's overall emotion and preferences related to music.

- Web scraping: The system utilizes a web scraper to extract tweets from users' public Twitter accounts.
- Music dataset: The system uses a music dataset to provide recommendations based on user emotion.
- Recommendation: The system uses machine learning algorithms to analyze user data and provide personalized recommendations.
- User feedback: The system allows users to provide feedback on recommendations to improve future recommendations.

3.2. Architecture Choices

- Web scrapers can access more data than the Twitter API. While the Twitter API provides access to recent tweets, a web scraper can scrape older tweets, as well as additional data such as user profiles, follower lists, and more. Also, web scrapers allow greater flexibility and customizability in terms of the data that is collected. With a web scraper, you can tailor your data collection to specific needs and parameters, whereas the Twitter API provides a more limited set of options.

- GloVe is a word embedding technique that captures global co-occurrence statistics to create word representations. It has several advantages, including capturing global context, producing linear relationships between words, being scalable and computationally efficient, offering pre-trained models that can be fine-tuned for specific tasks, and performing well on a range of NLP tasks. Overall, GloVe is a powerful and widely used technique for generating word embeddings that performs well on many NLP tasks.

- USE (Universal Sentence Encoder) is a pre-trained neural network that generates high-quality sentence embeddings. Sentence embeddings are numerical representations of the meaning of a sentence. USE can be used to calculate the similarity between sentences and the emotions with their patterns.

- DCNNs can automatically learn hierarchical representations of raw text data, which enables them to capture complex patterns in text and achieve state-of-the-art performance on sentiment analysis tasks. They can be trained on large datasets of labeled emotion data and can learn to recognise patterns in the data that are indicative of different emotions. This can then be used to predict the emotion of new unseen data.

- Content-based recommendation systems that use machine learning algorithms to analyze the features of items provide several advantages over collaborative filtering. They do not require user data, which can be a privacy concern, and they do not suffer from the cold-start problem. Content-based systems can provide highly personalized recommendations based on the user's past interactions with items and their attributes and features. They can also recommend items that the user might not have encountered, providing greater serendipity.

3.3. Constraints, Assumptions and Dependencies

Assumptions:

- The system assumes that users have public Twitter accounts and have tweeted quite a number of tweets to get a better understanding of their personality.
- The web scraper provides reliable and consistent access to web data.
- The emotional classifier is accurate and can correctly identify the sentiment of the scraped data.
- The music dataset used for content-based filtering is comprehensive and up-to-date.
- Users will provide honest and accurate feedback for collaborative filtering.

Constraints:

- The system is limited to the availability and quality of the data obtained from the web scraper and the music dataset.
- The performance of the emotional classifier and recommendation system may be affected by the size and complexity of the datasets.
- The user interface and functionality may be limited by the capabilities of the web application framework used for development.

Dependencies:

- The system relies on access to the web data obtained by the web scraper and the music dataset.
- The sentiment analysis algorithm and recommendation system depend on the quality and completeness of the datasets.
- The music that is recommended is dependent on the sentiment classified by the emotion classifier.

Interoperability requirements:

- The rest of the system must be able to interact with the web scraper to obtain and process web data.
- The emotional classifier system must be able to interface with the music dataset to perform content-based filtering.

Interface/protocol requirements:

- The system must be able to use the appropriate protocols for accessing the web scraper and the music dataset.
- The application's user-interface should be user-friendly and intuitive, allowing for easy navigation and interaction with the system.

Data repository and distribution requirements:

- The system requires a reliable and secure database to store user data and feedback, as well as the twitter and music datasets.

- The system should be able to efficiently distribute recommendations to users through the application.

Performance related issues:

- The size and complexity of the datasets may impact the performance of the sentiment analysis and recommendation algorithms.
- The system must be able to handle a large volume of data processing without significant latency or downtime.

End-user environment:

- Simple and intuitive interface: The interface should be designed to be as simple and intuitive as possible, with clear and easy-to-use navigation.
- The interface should also be responsive and adaptable to different screen sizes and resolutions.

Availability of resources:

- The system requires access to reliable and secure servers and storage to host the application and store user data.
- The system may require additional resources, such as computational power or memory, to handle large volumes of data processing.

Hardware or software environment:

- The system should be designed to work within the limitations of the hardware and software environment where it is deployed, including operating systems, browser versions, and device capabilities.

Deployment, maintainability, scalability, availability:

- The system should be easy to deploy on various platforms, such as cloud infrastructure or on-premises servers.
- The system should be scalable, allowing for increased capacity and usage as needed.
- The system should be highly available, minimizing downtime and ensuring reliability for users.

4. High Level System Design

Logical user groups:

- Registered Users: Users who have created an account and have access to all the features of the application such as rating songs, viewing recommendations, and accessing their user history.
- Administrators: Users who have administrative privileges and can manage the application, including managing user accounts and data, managing music data, and updating the application.

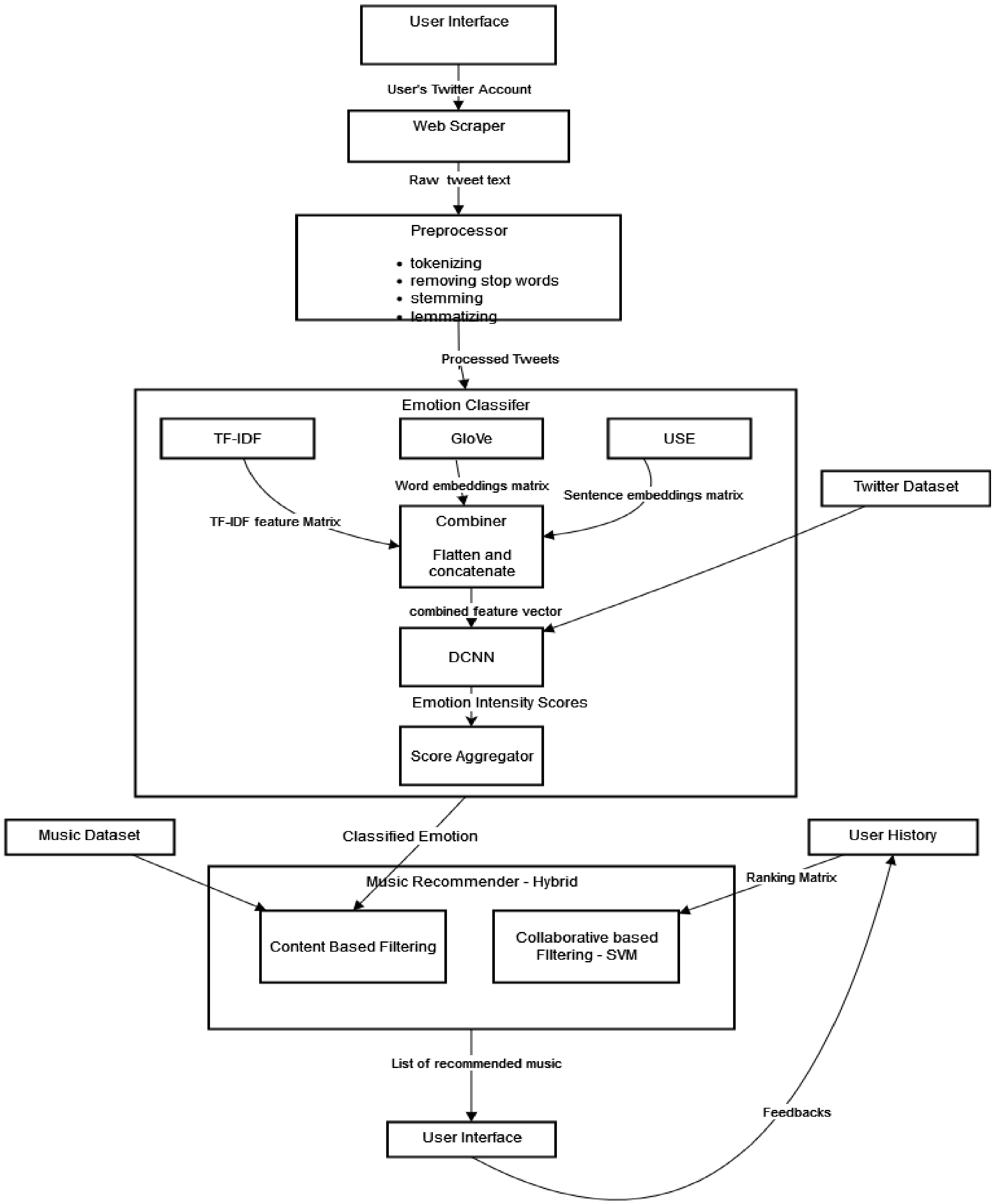
Application components:

- Web scraper: responsible for scraping data from Twitter.

- Data processing and storage: responsible for processing the scraped data, storing it in a database, and preparing it for analysis.
- Emotion Classifier: This component will use natural language processing techniques and machine learning to classify tweets into different emotions.
- Content-based filtering algorithm: responsible for recommending music based on emotion and the features of a given song.
- Collaborative filtering algorithm: responsible for recommending music based on the listening history and ratings of users.
- User interface: responsible for displaying the recommendations to the user, allowing them to rate songs, and providing a search function to find specific songs or artists.

Data components:

- User data: This would contain information about users such as their name, email, user ID, and user history.
- Tweet data: This would include data retrieved from Twitter such as tweet ID, tweet content, and the user who posted the tweet.
- Music data: This would contain data about the music such as artist name, track name, album name, release year, and genre.
- Rating data: This would include user ratings for various songs, such as the song ID, the user ID, and the rating given by the user.
- Training data: This would include the data used to train the machine learning models such as tf-idf, GloVe, USE, and DCNN.
- Recommendation data: This would be the data generated by the recommendation algorithm, which would contain the recommended songs and their associated scores.


User Interface
User's Twitter Account
Web Scraper
Raw tweet text
Preprocessor
• tokenizing
• removing stop words
• stemming
• lemmatizing
Processed Tweets
Emotion Classifer
TF-IDF
GloVe
USE
Word embeddings matrix
Sentence embeddings matrix
Twitter Dataset
TF-IDF feature Matrix
Combiner
Flatten and concatenate
combined feature vector
DCNN
Emotion Intensity Scores
Score Aggregator
Music Dataset
Classified Emotion
User History
Music Recommender - Hybrid
Ranking Matrix
Content Based Filtering
Collaborative based Filtering - SVM
List of recommended music
Feedbacks
User Interface

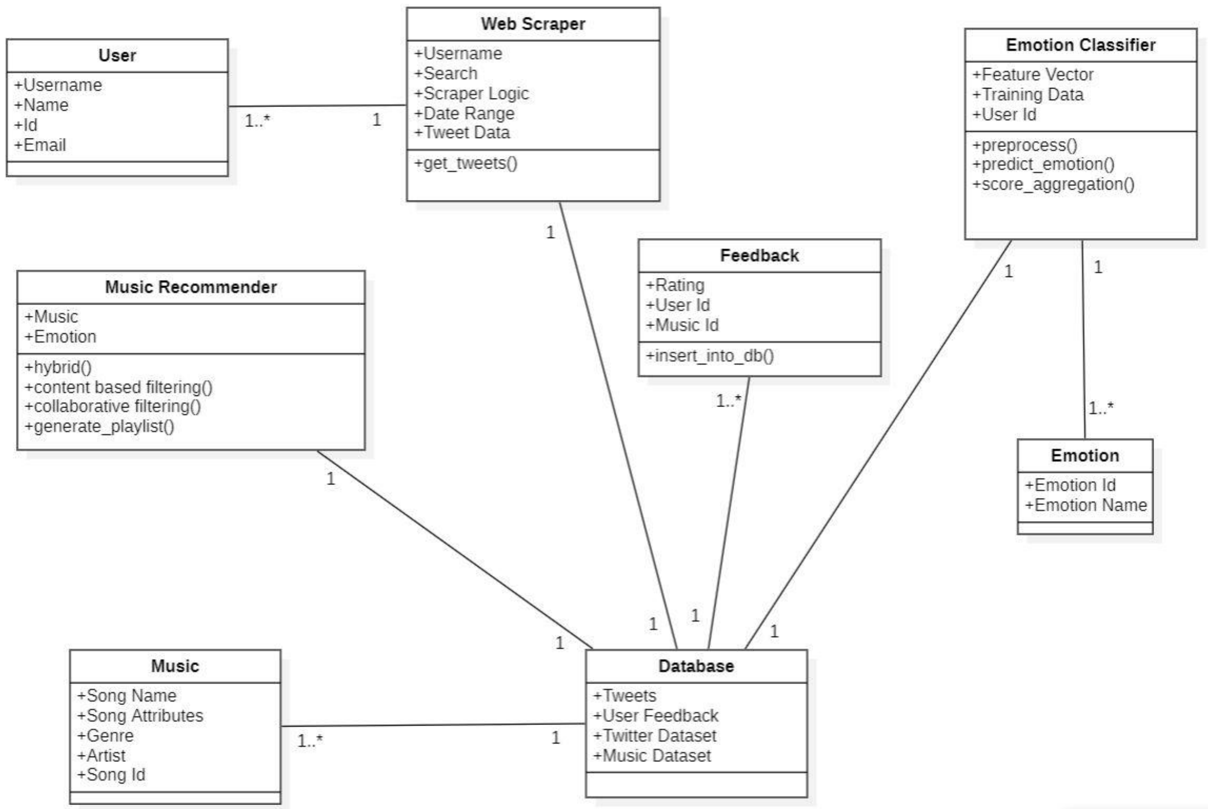
1. Module
   Project Management:
   - Defining project scope and objectives
   - Identifying project stakeholders and their roles
   - Developing a project plan that includes timelines, milestones, and resource requirements
   - Assigning tasks to team members and monitoring progress
   - Managing risks and issues that may arise during the project
   - Ensuring effective communication between team members and stakeholders
   - Conducting regular reviews and evaluations of project performance

   Code organization:
   - Defining coding standards and guidelines for the project
   - Establishing a clear and consistent naming convention for functions, variables, and files
   - Creating a modular code structure that separates different functionalities into modules or classes
   - Maintaining a consistent file and directory structure for the project
   - Ensuring that the code is well-documented, including comments in the code and a user manual or technical manual
   - Conducting regular code reviews and evaluations to ensure that coding standards and guidelines are being followed

2. Security
   - Authentication: Users should be required to log in with a username and password to access the system, and their credentials should be verified to ensure they are authorized to use the system.
   - Authorization: Once a user is authenticated, the system should have mechanisms in place to ensure they only have access to the parts of the system they are authorized to use.
   - Encryption: Sensitive data like user credentials, ratings, and user histories should be encrypted to prevent unauthorized access.
   - Access controls: Access controls should be in place to restrict access to the system's resources and data to only authorized users.
   - Regular updates and maintenance: Regular updates and maintenance of the system can help to ensure that security vulnerabilities are addressed in a timely manner.

5. Design Description

5.1. Master Class Diagram

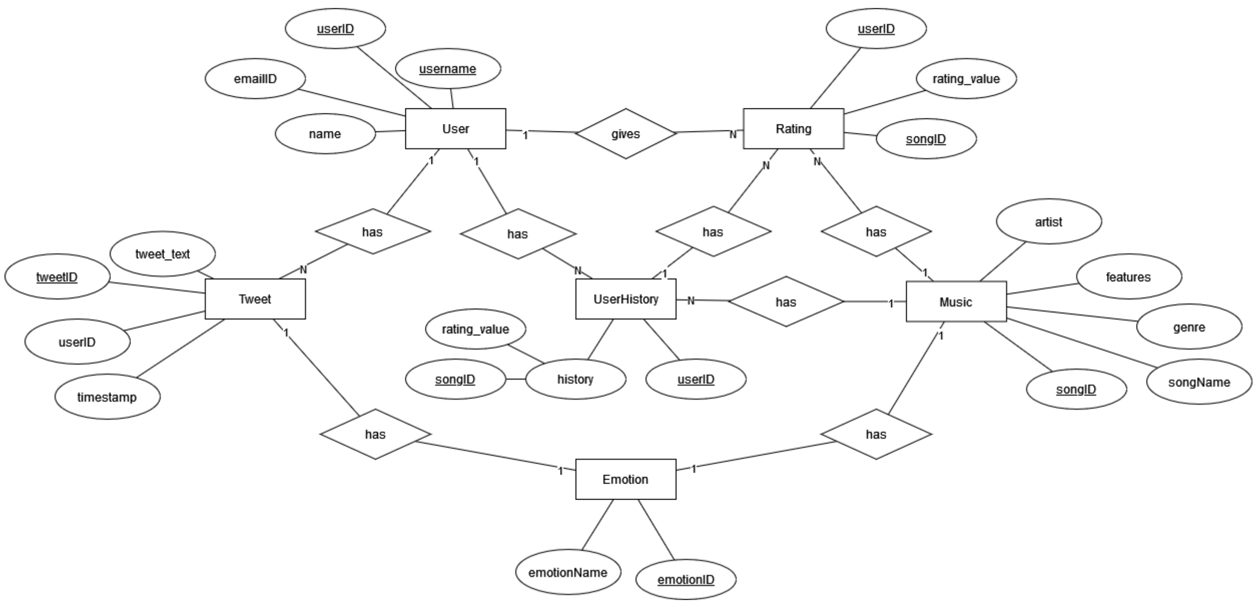


5.2. Reusability Considerations

The reusability considerations that can be taken into account to ensure that the project is easily maintainable and scalable are:

- Use of existing libraries and frameworks: To reduce development time and effort, existing libraries and frameworks can be utilized for tasks such as web scraping, sentiment analysis, and recommendation systems.
- Modular design: The project can be designed with modularity in mind, so that individual components can be easily reused in other projects. For example, the sentiment analysis module can be separated from the recommendation engine and reused in other sentiment analysis projects.

6. ER Diagram



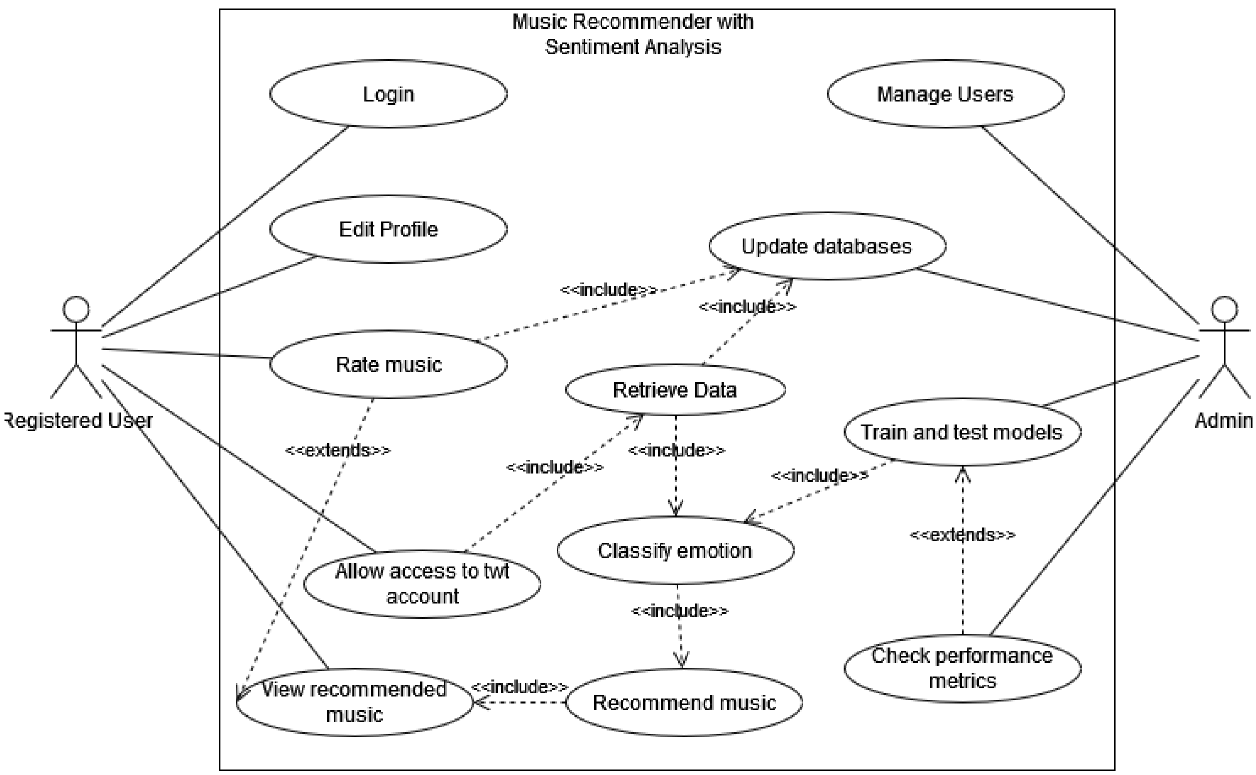
#	Entity	Name	Definition	Type
ENTITIES				
1.	User	User	A person who provides their twitter account to get music recommendations	Strong
2.	Tweet	User's Tweet	A tweet made by a user which is taken into consideration to find their emotion	Strong
3.	Emotion	Tweet's Emotion	The emotion analyzed from a user's tweet	Strong
4.	Music	Music	A song that is recommended based on the user's emotion	Strong
5.	Rating	Ratings	The feedback given by user for a song as ordinal ratings	Strong
6.	UserHistory	User's History	A record of user's ratings for each song	Strong

#	Attribute	Name	Definition	Type
DATA ELEMENTS				
1.	userID	User's ID	Unique identifier for each user	integer
2.	username	Twitter username	A user's unique twitter username	string
3.	name	User's Name	A user's full name	string
4.	emailID	User's email ID	A user's email that is sued for their twitter account	string
5.	tweet_id	Tweet's ID	Unique identifier for each tweet	integer
6.	tweet_text	Tweet content	The textual contents of the tweet	string
7.	timestamp	Tweet's timestamp	The time at which a tweet was posted	time
8.	emotionID	Emotion's ID	Unique identifier for each emotion	integer
9.	emotionName	Emotion	Emotions used for classification	string
10.	songID	Song's ID	Unique identifier for each song	integer
11.	songName	Song's name	A song's title	string
12.	genre	Song's genre	A song's genre	string
13.	features	Song's features	All the features of a song used to filter music	integer
14.	artist	Song's artist	The artist who sang the song	string
15.	rating_value	User's rating	The user's ordinal rating for a song	integer
16.	history	User's history	A user's rating for a song that was recommended previously	-

7. User Interface Diagrams

- Home Page: This is the first page that the user sees when they access the system. It can have options for signing up, logging in, and accessing various features of the system.
- Sign Up Page: This page allows new users to create an account on the system. It can have fields for entering personal information such as name, email, and password.
- Login Page: This page allows existing users to log in to the system. It can have fields for entering the email and password associated with the account.
- Music Recommendation Page: This page shows the music recommendations generated by the system. It can display information such as the song title, artist name, and a link to listen to the song.

- Music Details Page: This page provides more information about a particular song, such as the genre, album, and release date. It can also show the song lyrics, reviews, and related songs.
- User Profile Page: This page shows the user's profile information, such as their name, email, and music preferences. It can also allow the user to edit their profile and view their listening history.
- Search Page: This page allows the user to search for music by keyword, artist name, genre, or other criteria. It can display the search results and allow the user to listen to the songs and add them to their playlist



8. Report Layouts

- Introduction: This section introduces the project, its goals and objectives, and give an overview of the methodology used.
- Data Collection: This section describes the data sources and the process used to collect and clean the data.
- Exploratory Data Analysis: This section presents the descriptive statistics, visualizations, and other analyses used to understand the data and identify any patterns or relationships.
- Modeling: This section describes the machine learning models used for classification as well as recommendation: TF-IDF, GloVe, USE, DCNN, the collaborative filtering and content-based filtering algorithms.
- Data Storage: This section describes the four databases, one to store a user's tweets that serve as inputs, twitter dataset, music dataset, and a user's history.
- Results: This section presents the evaluation metrics used to measure the performance of the models, as well as the actual recommendations generated by the system.
- Discussion: This section provides an interpretation of the results and discusses the strengths and limitations of the models used.
- Conclusion: This section summarizes the findings of the project and provides recommendations for future work.
- References: This section includes a list of the sources cited in the report.

9. Help

Required manuals will be made available to the users and admins to aid in the usage of the system including:

The user manual: provides step-by-step instructions on how to use the music recommendation system. It would include information on how to register for an account, how to search for music, how to rate music, and how to access recommended music based on the user's preferences.

The technical manual: provides detailed information on how the system works and how it was developed. It would include technical specifications, such as the programming languages and frameworks used, as well as a description of the data sources and algorithms used for music recommendation. It would also include instructions for installing and configuring the system, as well as troubleshooting tips and suggestions for future development.

The troubleshooting manual: provides details on how the user can troubleshoot the system if or when they encounter any errors, such as the scraper being unable to retrieve tweets from the provided twitter account, the system is unable to classify emotion and recommend the personalized music playlist. It will also provide some general troubleshooting tips that users can follow if they encounter problems that are not covered in the guide.

10. Design Details

A music recommender system with sentiment analysis will rely on a combination of platforms, systems, and processes to function properly. Some of the key areas:

- Web scraping: The system will rely on web scraping to extract tweets related to music. We can perform this using python libraries or web scraping tools.
- Emotional Classifier: The system will use an emotional classifier composed of TF-IDF, GloVE, USE and DCNN to determine the emotion of each tweet. The overall emotion of a user is then found by using an aggregator.
- Recommendation engine: The system will need a recommendation engine that can take into account the emotion of each user and suggest appropriate music.
- Databases: The system will require databases to store user's tweets, the twitter dataset for training and testing the emotional classifier, the music dataset to recommend music and to store user's history.

In terms of changes, some of the key areas are:

- Novelty: Music Recommender systems that are currently available do not use sentiment analysis. Therefore using sentiment analysis helps make the music playlist more personalized.
- Performance: The system should be designed to handle a large volume of tweets and user requests. This may require scaling up the system infrastructure or optimizing the code to improve performance.
- Security: The system should be designed to ensure the security and privacy of user data. This may require implementing encryption, access controls, and secure data storage.
- Reliability: The system should be designed to be reliable, with mechanisms in place to handle errors and recover from failures. This may require implementing backup and recovery strategies, monitoring tools, and automated testing.
- Maintainability: The system should be designed to be maintainable, with well-organized code, clear documentation, and automated testing. This will make it easier to modify and extend the system over time.
- Reusability: The system can be designed with reusability in mind, with modular components and clear documentation. This will allow other developers to reuse the system components in other projects.
- Resource utilization: The system should be designed to utilize resources efficiently, such as minimizing memory and CPU usage, to reduce operating costs and improve scalability.

  The effectiveness of the music recommendation system will rely on its ability to seamlessly integrate and balance the various platforms, systems, and processes involved. This must be done while keeping in mind critical factors such as novelty, innovativeness, interoperability, maintainability, and performance, all of which are crucial to the system's smooth operation and its ability to achieve its goals.

Appendix A: Definitions, Acronyms and Abbreviations

- API: Application Programming Interface
- DCNN: Dilated Convolution Neural Networks - deep learning models that use a convolution operation with increased gaps between the input feature maps to increase the receptive field without increasing the number of parameters
- GloVe: Global Vectors for Word Representation - an unsupervised learning algorithm for obtaining vector representations for words.
- TF-IDF: Term Frequency-Inverse Document Frequency - a measure that can quantify the importance or relevance of string representations in a document amongst a collection of documents
- USE: Universal Sentence Encoder - encodes text into high dimensional vectors that can be used for text classification, semantic similarity, clustering, and other natural language tasks

Appendix B: References

- Irvin Dongo, Yudith Cardinale, Ana Aguilera "A qualitative and quantitative comparison between Web scraping and API methods for Twitter credibility analysis" in International Journal of Web Information Systems © Emerald Publishing Limited 1744-0084 DOI 10.1108/IJWIS-03-2021-0037
- Project Management Institute. (2017). A guide to the Project Management Body of Knowledge (PMBOK guide) (6th ed.). Project Management Institute.
- P. K. K. N. and M. L. Gavrilova, "Latent Personality Traits Assessment From Social Network Activity Using Contextual Language Embedding," in IEEE Transactions on Computational Social Systems, vol. 9, no. 2, pp. 638-649, April 2022, doi: 10.1109/TCSS.2021.3108810.
- H. Ahmad, M. U. Asghar, M. Z. Asghar, A. Khan and A. H. Mosavi, "A Hybrid Deep Learning Technique for Personality Trait Classification From Text," in IEEE Access, vol. 9, pp. 146214-146232, 2021, doi: 10.1109/ACCESS.2021.3121791.
- Praphula Kumar Jain, Waris Quamer, Vijayalakshmi Saravanan, and Rajendra Pamula, “Employing BERT-DCNN with sentic knowledge base for social media sentiment analysis” in Journal of Ambient Intelligence and Humanized Computing, January 2022
- Saba Yousefian Jazi 1 & Marjan Kaedi1 & Afsaneh Fatemi1,”An emotion-aware music recommender system: bridging the user’s interaction and music recommendation” in Multimedia Tools and Applications(springer),15 January 2021

Appendix C: Record of Change History

#	Date	Document Version No.	Change Description	Reason for Change
1.				
2.				
3.				

Appendix D: Traceability Matrix

Project Requirement Specification Reference Section No. and Name.	DESIGN / HLD Reference Section No. and Name.
Reference Section No. 4, Functional Requirements	Reference Section 10 - Design Details
Reference Section No. 6 - Non Functional Requirements	Reference Section 10 - Design Details, Reference Section 3 - Design Considerations, Reference Section 4 - High Level System Design