EMOTION BASED MUSIC PLAYER

P.P.D.Srivalli

March 21, 2023

Abstract

The work describes the development of Emotion Based Music Player, which is a computer application meant for all type of users, specifically the music lovers. Most people will randomly play the songs in the playlist. As a result, some of the songs not matches the users current emotion. It is aimed to provide a better enjoyment to music lovers in music listening..

1 INTRODUCTION

This Software Requirement Specification (SRS) report expresses complete description about Recommendation System Project. This document includes all the functions and specifications with their explanations to solve related problems as a project of Information Technology Department.

1.1 Purpose

The purpose of this document is to present a detailed description of the Recommended System that we will design and implement. It will explain the purpose and features of the system, interfaces of the system and what the system will do.

The users may or may not be aware of their need for a recommendation feature on the software or website they are using, but such features can increase efficiency and save time to users.

1.2 Scope

The goal is to design a recommended system on music domain. The system will be composed of server components and client components.

The server components will manage the database operations and algorithms that produce recommendation results. The client components will be graphical interfaces that are integrated into corresponding larger systems.

Generally, a music recommended system consists of three key components: users, items and user-item matching algorithms.

1.3 Problem Definition

We are working on some problems that the recommended systems may face and trying to improve the current solutions on these problems to increase the accuracy of the recommendations.

The main issues would be huge, dense and sparse data sizes. There also may be issues about similarity, calculations, and data integration.

Sparsity of data is also a major issue, because no matter how well the algorithms are, if there is no data to process, the system will be of no use.

Therefore, we need to work heavily on different algorithms and databases to create an efficient solution even for very large data sets.

1.4 System Overview

The following section of this document will focus on describing the terms of product perspective and functions, user characteristics, assumptions and dependencies.

Type	Percentage	Features
Savants	7	Everything in life seems to be tied up with music. Their musical knowledge is very extensive.
Enthusiasts	21	Music is a key part of life but is also balanced by other interests.
Casuals	32	Music plays a welcome role, but other things are far more important.
Indiff erents	40	They would not lose much sleep if music ceased to exist, they are a predominant type of listeners of the whole population.

Table 1: User Listening Experience Data Categorization.

In the third section, We will address specific requirements of the system, which will enclose external interface requirements, functional requirements, performance requirements, and other requirements of the system.

1.5 References

1. "EMO PLAYER" : Emotion Based Music Player
   https://www.irjet.net/archives/V5/i4/IRJET-V5I41065.pdf
2. "IJARCCE" : International Journal of Advanced Research in Computer and Communication Engineering
   https://ijarcce.com/upload/2017/april-17/IJARCCE%20140.pdf
3. "SRS" Software Requirement Specification Of Music Recommendation System CENG HISTORY X
   https://senior.ceng.metu.edu.tr/2014/cenghistoryx/documents/CengHistoryX_SRS.pdf
4. Emotion Based Music Player by "TAN SIEW CHING"
   https://utpedia.utp.edu.my/6334/1/TanSiewChing12873_finaldissertation.pdf
5. Sanjay Ghodawat University (India) : Emotion Based Music Player
   https://core.ac.uk/download/pdf/211922219.pdf

2 OVERALL DESCRIPTION

In this part, background information about specific requirements of the system will be provided briefly. General issues that affect the product and outline of the functional requirements will be mentioned, too. In short, this section will mainly give information about product perspective, product functions, constraints, assumptions and dependencies.

2.1 Product Perspective

Recommendation system depends on data comes from users and company items. This program has different types of users, so functionality differences between users will occur with respect to the data.

The recommendation systems used by companies have differences for efficiency. It should work efficiently according to music data. So, there exists a user interface that is suitable for music recommendations.

Our system will be working in background. Once the recommendation system finds an accurate result, it will be shown on the interface.

In terms of software, our recommendation system will run on personal computers, smart phones etc. i.e., it will run on any device with internet connection.

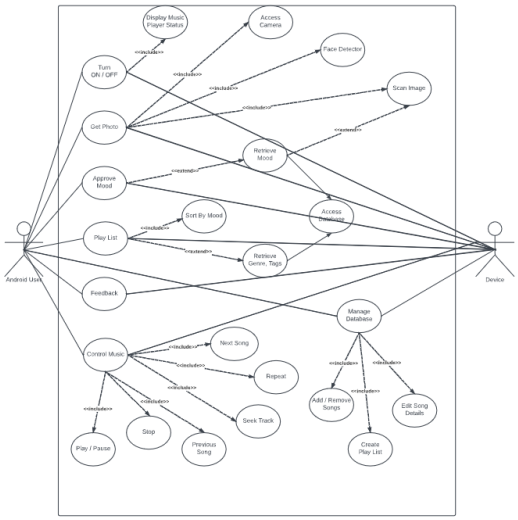

Figure 1: Use Case Diagram.

2.2 Product Functions

Use case diagram of the recommendation system and the other subsystems are revealed in above diagram. Steps are gathered in distinct entities, the functions of which are stated in further subsections.

2.3 User Characteristics

There will be one type of user. Therefore there are no differences between users in terms of functionality, visualization and interface. The user interface is only depend on the websites designers.

The website that our system is working on the background of it can offer different opportunities to its users. Then the recommendations will be specified according to the algorithm that depends on user actions.

These actions are determined by some parameters such as the time that the user listened the music. Some of these actions are listened, downloaded, listened before.

2.4 Constraints

- Since we need user profile data while developing the product, to find real time and sufficient data can be a problem for developer because of regulatory policies.
- Millions of data will be needed to test the software. At this stage developers will need huge amount of space and clusters.
- We will not design any specific interfaces for the product. So, developers have to consider common components of the applications so that the software can be integrated to any of the application easily.
- The application gathers real time user profile information from user accounts. Therefore, it must be reliable and keep those data in safe.
- Most important concern of the system is producing accurate recommendations. To provide expected accuracy and to handle huge data at the same time is critical.

2.5 Assumptions

As stated in the previous section which is constraints, there are several requirements like music data, user data, database management tool etc.

However, the case of having all these software and hardware provide, still we might have some difficulties to test the system without strong internet connection.

We stated that we will use Java as a programming language to code our system. However, we can also study in other platforms. For example, we might study in .NET Framework

2.6 Dependencies

To accomplish activating recommendation system, the requirements like music data, user data, database management tool etc should be provided.

Losing connection is an important problem, because our system will work on online platform. If the internet connection cannot be supplied as requested, it will fail and our system will be useless.

3 SPECIFIC REQUIREMENTS

This section will describe the software requirements in detail as subsections which are interface requirements, functional and non-functional requirements.

3.1 External Interfaces

Since there is just one type of user, the application will have only the user interface. Basically the user interface will direct the user through steps and will display the recommendations found by our algorithm. The steps of the user interface are the following :

3.1.1 Supplier System

This user interface is only for supplier, which is company. Supplier can do initialize service, analyze system performance, manage data sources, and modify recommendation algorithms operations.

3.1.2 User Subsystem

This user interface is for customers, that is the people who have an account or who want to be a member of the website. A user can do create an account, read recommendations (for user who has an account), log in operations.

3.1.3 Recommendation Subsystem

This user interface is the one in which our algorithm will work in the background and interface will be used by both customers and suppliers.

Customer cannot do many operations, but their feed-backs are very important to create a relevant recommendation.

Users can only use provided feedback operation. View recommendations operation can be used after our algorithm gives the result

3.2 Functions

In this section, we will explain the major functions of the Recommendation System

3.2.1 Client

The client of the system will be an application with a user interface that is integrated into a music listening website or application. This application gathers the information from users, investigates some actions of the users, and provides the connection with the server.

- Requesting Recommendations
- Display Recommendations
- Investigating User
- Evaluating Songs

SRS	Software Requirement Specification
Neo4j	Neo4j is a robust transactional property graph database.
Java	Java is a computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible.

Cypher	Cypher is a Graph Query Language.
Neoclipse	Neoclipse is a standalone workbench application to interact with Neo4j (database directory or server).
Windows	Windows is a series of graphical interface operating systems developed, marketed, and sold by Microsoft.
Netbeans	NetBeans is an integrated development environment (IDE) for developing primarily with Java, but also with other languages, in particular PHP, C/C++, and HTML5.
GUI	Graphical User Interface
Eclipse	A multi-language Integrated development environment (IDE)

3.2.2 Server

The server system will hold the entire data in a graph database, and must include all functionality to perform operations on this database, receive requests from the clients, evaluate, create and send recommendations etc

- Recommend using content based filtering
- Handle recommendation requests
- Store Evaluations
- Data Storing

3.3 Performance Requirements

- Accuracy :

  Since we will give the priority to the accuracy of the software, the performance of the Music Recommended will be based on its accuracy on recommendations.

- Failure Handling :

  System components may fail independently of others. Therefore, system components must be built so they can handle failure of other components they depend on.

- Openness :

  The system should be extensible to guarantee that it is useful for a reasonable period of time.

- Security :

  Sensitive information should be kept in safe.

3.4 Logical Database Requirements

This section describes information domain for the Music Recommender.

1. Data Description :

   Data objects in the Music Recommender that will be managed/manipulated by the software are described in this section with their attributes using class diagrams.

   - Data objects :

     Music recommendation system roughly has 5 types of data objects, namely User, Item , Recommendation, UserManager, EvaluationMeasures.

     (a) User :
     This object will hold the information of a specific user; id, name, age etc.

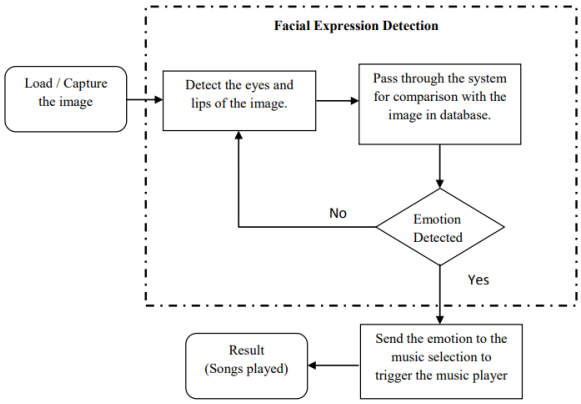

Figure 2: The Flow Chart of the Proposed Model.

(b) Item
This object will hold information about each specific music item; id, name, artist, album, genre etc.

(c) Recommendation :
This object will hold a specific rating value that a user gave to a item, including the required information of the user and item along with a rating value.

(d) User Manager :
This object handles the user addition and deletion operations.

(e) Evaluating Measures :
This object handles the operations about evaluating the data.

- Data Dictionary
  We have given details about description of object attributes at section 4.1.1. Therefore, it is not necessary to mention about them again.

3.5 Design Constraints

- Language :-
  The product should be integratable with any music websites. Also we will handle with a lot of parameter and object. At this point, we need an object-oriented and commonly used 17 language.
- Hardware Constraints :-
  The system will be integrated with a website. To use recommendation system, user should enter from a personal computer, mobile device with internet connection, tablet etc.
- Software System Attributes :-

1. Usability :
   The software will be embedded in a website. It should be scalable designed to be easily adopted by a system.
2. Reliability :
   The system should have accurate results and fast responses to user's changing habits.
3. Security :
   User profile information will be used, so data security is one of the most important concern of the system.

3.6 Key Features

Emotion based music player is an approach that helps the user to automatically play songs according to the emotions of the user. It recognizes the facial emotions of the user and plays the songs according to their emotion. The emotions are recognized using a machine learning method EMO algorithm.

4 VALIDATION CHECK

The validation of an Emotion Based Music Player can be assessed through several means:

4.1 Validity Check

1. User Feedback :
    - The most direct and immediate way to determine the effectiveness of an emotion-based music player is through user feedback.
    - Overall, the validity of an emotion-based music player depends on the accuracy of the algorithms used to analyze and interpret user emotional states, as well as the effectiveness of the music selection in regulating emotional states.

4.2 Consistency Check

Consistency is an important aspect and can be evaluated through various approaches such as test-retest reliability, inter-rater reliability, comparison to established emotion models, and user feedback.

- Test-retest reliability :

  This involves using the music player on multiple occasions and comparing the music selection to determine if the same emotional state consistently results in the same music selection.

- Inter-rater reliability:

  Multiple raters can independently evaluate the emotional state of a user and determine if they agree on the corresponding music selection.

- By checking consistency, the accuracy and reliability of the music player can be determined.

4.3 Realism Check

Realism refers to the degree to which an emotion-based music player accurately reflects the real world. The realism of an emotion-based music player is an essential aspect of its effectiveness.

- Ecological validity:

  Ecological validity refers to the extent to which the music player reflects real-world situations. If the music player is typically listened to experience a particular emotion in their daily lives, then the realism of the music player is high.

- Cultural validity:

  Cultural validity refers to the extent to which the music player reflects the cultural differences in music preferences for a given emotion.

  For example, different cultures may associate different types of music with happiness or sadness, and a realistic music player should reflect these cultural differences.

- By checking the ecological and cultural validity, the accuracy of the emotional recognition algorithms, and user feedback, the realism of the music player can be evaluated.

4.4 Verifiability Check

Verifiability is an important aspect and by checking the transparency of algorithms, replicability of results, independent evaluation, and peer review, the verifiability of an emotion-based music player can be evaluated.

- Independent evaluation :

  The music player should be evaluated by independent experts in the field, such as music therapists. These experts can evaluate the accuracy of the algorithms and the effectiveness of the music selection in regulating emotional states.

- Transparency of algorithms :

  The algorithms used to analyze and interpret user emotional states should be transparent. This means that the programming code and underlying data should be available for review by independent experts.

  The more transparent and open the algorithms and data are, the more verifiable the music player is likely to be.

4.5 Completeness Check

Completeness refers to the music player which covers all possible emotional states and provides appropriate music for each emotions.

Completeness can be evaluated through various approaches such as inclusion of a broad range of emotional states, adequate coverage of music genres, user feedback, and expert evaluation.

- Broad range of Emotional States:

  The music player should include a broad range of emotional states to ensure that it can effectively regulate a variety of emotional states.

  This can be achieved by using established emotion models that include a broad range of emotional states.

- Expert evaluation:

  The music player can be evaluated by experts in the field, such as music therapists, to determine if the music selection is appropriate for each emotional state.

- By checking completeness, the effectiveness and usefulness of the music player can be determined.

5 VALIDATION TECHNIQUES

5.1 Requirement Reviews

Requirement reviews are an important part of developing an emotion-based music player.By conducting requirement reviews, the design and features can be evaluated.

To ensure that it meets the needs of its intended users, can function effectively, is compatible with a variety of devices and platforms, and has appropriate security and privacy features.

- User requirements :

  The user requirements should be evaluated to ensure that the music player meets the needs of its intended users. This can involve conducting user surveys, focus groups, or user testing to identify the specific requirements of the target audience.

- Technical requirements :

  The technical requirements should be evaluated to ensure that the music player can function effectively. This can involve assessing the hardware and software requirements of the music player, as well as the algorithms used to recognize and respond to emotional states.

- Security and privacy requirements :

  The security and privacy requirements should be evaluated to ensure that user data is protected. This can involve assessing the security and privacy features of the music player, as well as testing the music player for vulnerabilities and potential data breaches.

5.2 Prototyping Techniques

Prototyping is an important step in the development of an emotion-based music player and it involves creating a working model of the music player to test its functionality and user experience.

- Low-fidelity prototypes:

  Low-fidelity prototypes are simple and quick to create, using basic materials like paper, cardboard, or post-it notes. They can be used to test and refine the user interface design, features, and functionality of the music player.

- High-fidelity prototypes:

  High-fidelity prototypes are more advanced than low-fidelity prototypes, using advanced materials like digital wire-framing tools, interactive software, or hardware components. They can be used to test and refine the user interface design, features, and functionality of the music player in a more realistic environment.

- By using these prototyping techniques, developers can create a working model of the emotion-based music player, test its functionality and user experience, and gather feedback from users to refine its design and features.

5.3 Test case Generation

Test case generation is an important step in the development of an emotion-based music player. Test cases are designed to evaluate the functionality and effectiveness of the music player in recognizing and responding to different emotional states.

- Develop test cases :

  Based on the defined emotional states, music selection criteria, testing scenarios, and testing environment, test cases can be developed to evaluate the functionality and effectiveness of the music player in recognizing and responding to emotional states.

- Test execution :

  Once the test cases are developed, they can be executed to evaluate the music player's performance in recognizing and responding to emotional states.

- Test evaluation :

  The results of the test execution should be evaluated to determine the music player's effectiveness in recognizing and responding to emotional states. The evaluation should identify any issues or areas for improvement.

- Test cases can be generated to evaluate the effectiveness of the music player in recognizing and responding to emotional states.

  This process can help developers improve the design and functionality of the music player, ensuring that it effectively meets the needs of its users.