

Mine-Music: A Music Player Based on Data Mining Technology

Atrima Ghosh

B-Tech, Electronics and Communication Engineering, IEM, WBUT, India

Abstract- This report describes the concept and implementation of a system which allows users to play music as per their preferences and provides the users with recommendations as well from the system after log in. The data mining technology has been utilized in the making of this system in order to rate the users' likes and choices, taking into consideration the similarity in the backgrounds and personal information of the users obtained during account creation. When a new user creates an account in the system and chooses to listen to a song for the first time, the system automatically provides him or her with a number of initial options based on the data already present in the database, which have been specifically sorted out on the basis of the information provided by the user during the creation of his or her. From the options available, the system further ranks the songs according to their popularity among users sharing the same background. However, if an existing user logs in and chooses to listen to a song, then the system provides him or her with options based on his previous records, taking into account the songs he or she has listened to in the hour which matches the hour of the user's log in time. Furthermore, the system ranks the filtered options based on the number of times the user has listened to them. If the user finds what he is looking for among the displayed options itself, then he can select the respective option. If not, then he can search for his choice in the database. If available, then the user can play the song and if not, then the system asks him or her to enter further details regarding the song, which is saved in the database so that it can later be used to fetch the song from other sources and thereby make it available to the users.

Keywords: Data mining, recommendation system, python, Graphical User Interface, SQL.

I. MOTIVATION

The acceptance of a system mainly depends on its feasibility, its efficiency and its compatibility with the user. We have many music players already available which instantly play songs chosen by the user. However, I wanted to create a system which not only plays songs but also recommends a few appropriate ones for the corresponding user, based on certain parameters like nationality, language, age or the time of the day when the user has logged in. Using this recommendation technique makes the system much more user friendly since it makes it much easier for the users to access their favorite songs. It also ensures that a user requires much less time in order to play a song of his or her choice. In addition to it, I have used a popularity index parameter for all the songs which aids in keeping the user updated

regarding songs that are highly popular among other users who share similar interests. Also, if an existing old user logs in, the system notes the log in time and displays the names of mainly those songs which the user has listened to maximum number of times during that same hour of the day, thereby saving much time which would otherwise be required search for the song. Thus, the system helps the user in many ways to make it easy for him or her to listen to a song by providing options, based on the statistics stored in the database, most likely to match his or her choice.

II. INTRODUCTION

The system utilizes the following features for efficient performance:

2.1 PYTHON PROGRAMMING LANGUAGE

Python is an interpreted, high-level programming language, pure object-oriented, and powerful server-side scripting language for the Web. Like all scripting languages, Python code resembles pseudo code. Its syntax's rules and elegant design make it readable even among multiprogrammer development teams. The advantages of using python as a programming language include:

- syntax is clear and readable. It is excellent for beginners, yet superb for experts.
- self adjustable. Python provides full dynamic run-time type checking and bounds checking on array subscripts. It employs garbage collection so there is no problem with dangling pointers or memory leaks.
- supports object oriented programming.
- runs everywhere. Python programs are platform independent, which means that regardless of the operating system, it will run on any other computer with Python.
- open source. In addition to python being free, its license also allows us to modify python.
- highly scalable, suitable for large projects as well as small ones
- portable, cross-platform
- embeddable
- easily extensible
- powerful standard library

2.2 STRUCTURED QUERY LANGUAGE

SQL (Structured Query Language) is a special-purpose programming language designed for managing data held in a relational database management system.

Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language and a data manipulation language. The advantages of using SQL include:

- High speed. SQL Queries can be used to retrieve large amounts of records from a database quickly and efficiently.
- SQL makes it easier to manage database systems without having to write substantial amount of code.
- The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control.

2.3 GRAPHICAL USER INTERFACE

In computing, graphical user interface (GUI) is a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces, typed command labels or text navigation. The actions in GUI are usually performed through direct manipulation of the graphical elements. The advantages of using a graphical user interface include:

- User friendly. GUI systems offer a comparatively intuitive approach. Even users without any significant training can learn to use the systems with ease and hence use softwares to accomplish their goals.
- Multitasking. GUI systems offer a simple means of multitasking. Users can maintain multiple open applications and transition between them with a click of the mouse. This enhances productivity.
- Visuals. GUI systems provide a more pleasant visual environment to work with on the computer, which makes GUI a desirable feature for most end-users.

2.4 DATA MINING

Data mining is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. Aside from the raw analysis step, it involves database and data management aspects, data pre-processing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of discovered structures, visualization, and online updating. Data mining uses information from past data to analyze the outcome of a particular problem or situation that may arise. It is extremely user friendly, reduces access time to data and also allows us to filter or extract only a small portion of required data from a huge database.

Step 4: Select login.

Step 5: Open a new window where the user enters username and password and selects OK.

Step 6: Check whether the username and password are a match with those present in the database.

Step 7: If username and password are valid, go to step 8. Else enter correct username and password.

Step 8: Login successful.

Step 9: Note down the time of logging in.

Step 10: Check in the database to see if the user has listened to any song in the past at that particular hour of the day. If so, then go to step 11, else go to step 13.

Step 11: Check to see the number of times the user has listened to the songs and accordingly arrange the list of recommended songs, keeping the most listened to songs at the top.

Step 12: If the total number of recommendations are fulfilled, then go to step 15. Else, for the selection of remaining songs, go to step 13.

Step 13: Check the main Songs database to see which songs are most listened to by the users having the same nationality, age group and language as that of the logged in user.

Step 14: Out of the selected songs, select the ones with the highest popularity index and arrange them accordingly.

Step 15: Display the entire list of recommended songs for the user to choose. If the user selects from among them, then continue. Else go to step 18.

Step 16: Play the song.

Step 17: Update the database which keeps track of the users' past records of the songs which the users listen to. Go to step 26.

Step 18: If the desired song is not displayed, then ask the user to enter the name of the song that he wants to listen to.

Step 19: If available, go to step 16. Else continue.

Step 20: Ask the user to enter the details of the song – Song name, Artist, Album name, Year released, Genre and language of the song.

Step 21: Store the details in the database for unknown songs. Go to step 26.

Step 22: Select Sign up.

Step 23: Enter the details – Name, Address, Country, Language, Date of Birth, Username and Password.

Step 24: Check to see if the username already exists in the database. If not, then continue. Else, re-enter username.

Step 25: The account is successfully created. Go to step 4.

Step 26: Stop.

III. METHODOLOGY

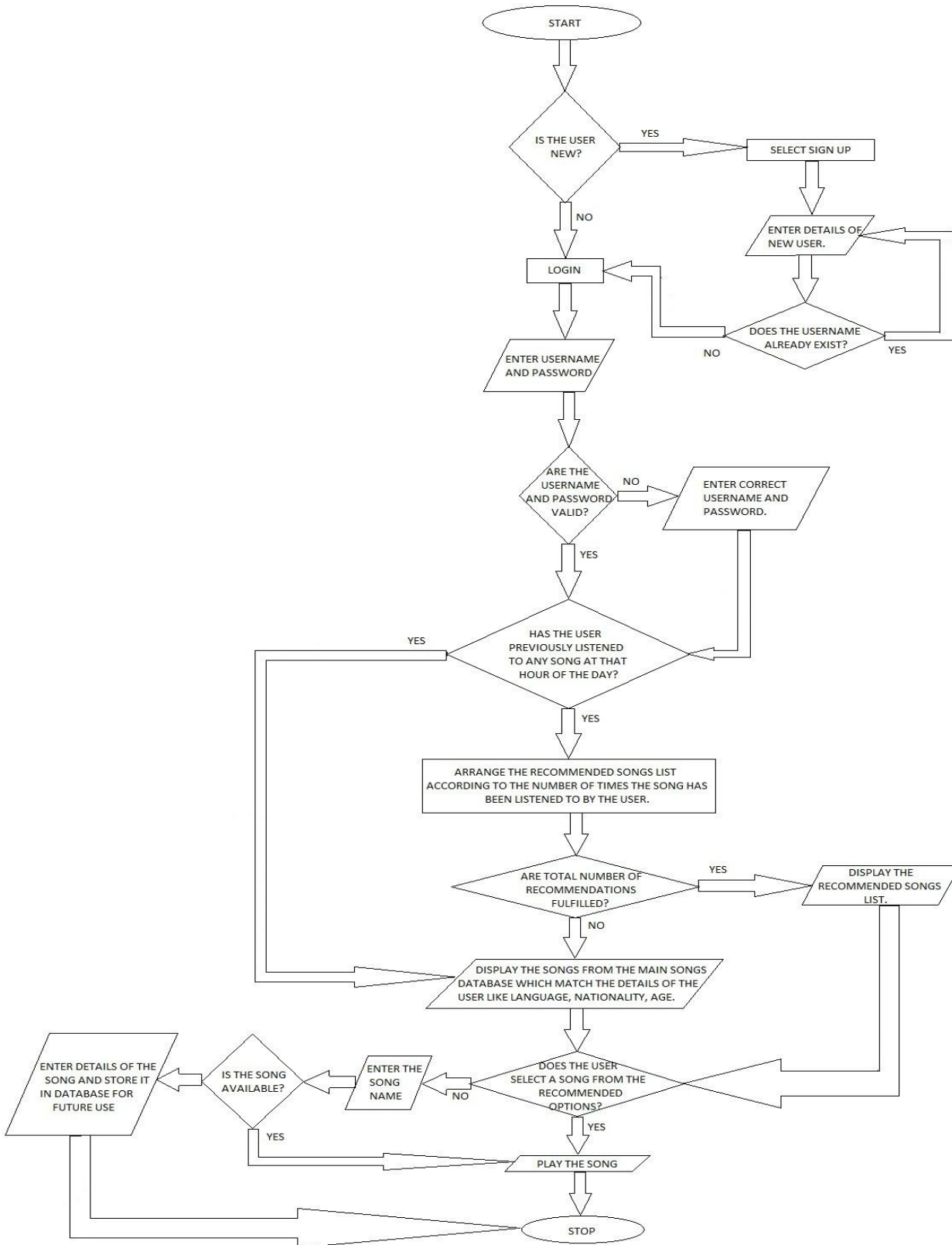
3.1 STEP ALGORITHM:

Step 1: Start.

Step 2: Display a window for old users to log in and new users to sign up.

Step 3: If new user, go to step 22, else continue.

3.2 FLOW CHART:



3.3 DATABASES:

- User Accounts database:
Name, Address, Country, Language, DateOfBirth, Username, Password.
- Songs database:
SongID, SongName, Artist, Genre, AgeLow, AgeHigh, Language, Country, Time, PopularityIndex.
- Preference database:
Username, SongID, Time, Count.
- Unknown Songs database:
SongName, Artist, Album, YearReleased, Genre, Language.

IV. RESULTS

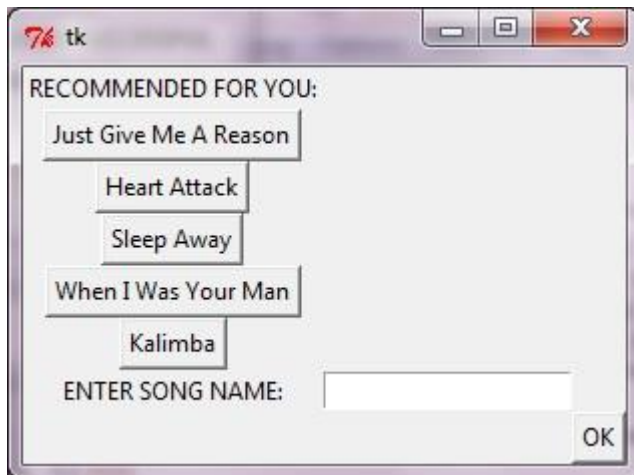


Fig. 1.

The above screenshot shows the songs recommended by the system in case of a new user who has logged in for the first time.



Fig. 2.

After logging in a number of times and listening to multiple songs, the system suggests the above songs. There has clearly been a change in the order as well as the name of the recommended songs, since the first time the user had logged in.

V. CONCLUSION AND FUTURE WORK

Thus, we see that the system proposed can be made to work efficiently. The methodology section describes how the entire process can be implemented and how the entire system can be developed to achieve our goal. The screenshots provided in the results section show that the system is working.

The system can be further developed in order to add security features to it. The system created is vulnerable because no measures have been taken to make the overall process secure. Thus, we need to import security measures in order to protect the system. Also we can try to import better data mining algorithms so as to make the system much more efficient both in terms of time as well as cost. As a whole, we hope the implementation process described will aid the developer's community in near future.

AUTHORS

First Author – Atrima Ghosh, atrimaghosh@gmail.com, +91-9433226191, B-Tech, Electronics and Communication Engineering, IEM, WBUT, India