A Recommender System based on genetic algorithm for music data search

Project Report

Walchand Collegeof Engineering, Sangli.

Department of Computer Science and Engineering

TEAM : Aspirants

Name	Exam Seat No	Roll No
Boob Nikhil Dinesh	2011BIT008	105
ChandakRajatGovind	2011BCS012	12
Chougule Deepak Sanjay	2011BCS015	15

Project Guide

Dr. S.H.Bhandari

Academic Year

2013-2014

Department of Computer Science and Engineering
Walchand College of Engineering, Sangli

CERTIFICATE

This is to certify that the Third year B.Tech. Project entitled

A Recommender System based on genetic algorithm for music data search

Submitted by

Mr. Nikhil Boob
Mr.Rajat Chandak
Mr.Deepak Chougule

For the partial fulfilment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering at Walchand College of Engineering, Sangli is a bona fide work carried out during academic year 2013-14.

Head,
Department of C.S.E.

Project Guide

Examiner(s)

1._____

2._____

Date:
Place:

Table of Contents

1.0 Introduction:

1.1 Purpose:

This document aims to give a brief description about the **Recommender System Based on Genetic Algorithm for Musical Data**.

The issue of a recommender system is how to recommend items with user's preferences from resources. Initially it was collaborative filtering approach, system provides recommendation by collecting user's profile and discovers relations between each profile and according to match for the profile system was used to recommend items.

But , In this system we are using content based filtering approach in which along with user preferences , description of the items are also considered for the recommendation of the items[1].

1.2 Objectives:

- To study and implement genetic algorithm efficiently for music data search[2].
- To accurately recognize the trend of user's preferences and provide him adaptive recommendation for music data.

1.3Scope:

- Users can rate the songs according to their individual choices and get preferences.
- User can hear song, and rate accordingly.

1.4Definitions, Acronyms, and Abbreviations:

- I-GA(Interactive Genetic Algorithm)
- CBF(Content Based Filtering)
- JDK(Java Development Kit)

1.5Technologies to be used:

- **JAVA (JDK 1.5.0):**Development Language.
- **W3C package:**For parsing of the file.
- **CLAM Music Annotator:** For extraction of music features.

2.0Project Planning

2.1Project Planning and Scheduling

Project planning is part of project management, which relates to the use of schedules such as Gantt charts to plan and subsequently report progress within the project environment. Initially, the project scope is defined and the appropriate methods for completing the project are determined. Following this step, the durations for the various tasks necessary to complete the work are listed and grouped into a work breakdown structure. The logical dependencies between tasks are defined using an activity network diagram that enables identification of the critical path. Float or slack time in the schedule can be calculated using project management software. Then the necessary resources can be estimated and costs for each activity can be allocated to each resource, giving the total project cost. At this stage, the project plan may be optimized to achieve the appropriate balance between resource usage and project duration to comply with the project objectives. Once established and agreed, the plan becomes what is known as the baseline. Progress will be measured against the baseline throughout the life of the project. Analyzing progress compared to the baseline is known as earned value management.

2.2 Project Development Approach

We have used Iterative and Incremental Development model (IID) for our projectdevelopment. This development approach is also referred to as Iterative WaterfallDevelopment approach. Iterative and Incremental Development is a software developmentprocess developed in response to the more traditional waterfall model.

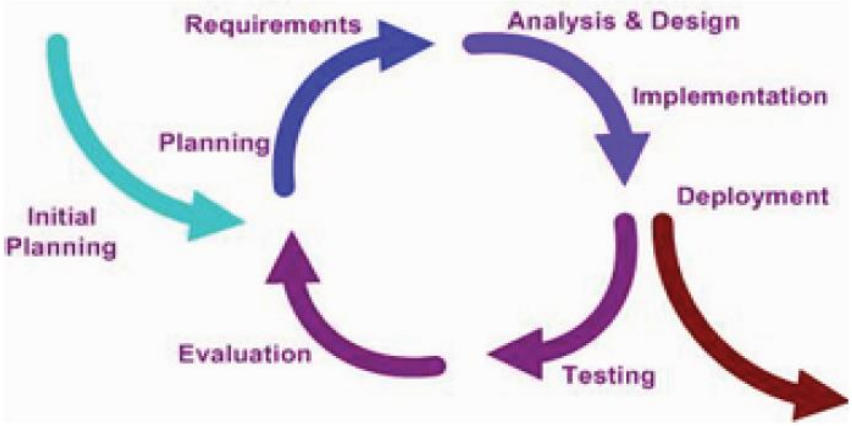


Fig 2.2.1: Iterative and Incremental Life cycles

The basic idea behind iterative enhancement is to develop a software systemincrementally, allowing the developer to take advantage of what was being learned

duringthe development of earlier, incremental, deliverable versions of the system. Learningcomes from both the development and use of the system, where possible. Key stepsin the process were to start with a simple implementation of a subset of the softwarerequirements and iteratively enhance the evolving sequence of versions until the fullsystem is implemented.

At each iteration, the procedure itself consists of the Initialization step, theIteration step, and the Project Control List as shown in *Fig 2.2.1*. The initialization step creates a base versionof the system. The goal for this initial implementation is to create a product to which theuser can react. It should offer a sampling of the key aspects of the problem and provide asolution that is simple enough to understand and implement easily. To guide the iterationprocess, a project control list is created that contains a record of all tasks that need to beperformed. It includes such items as new features to be implemented and areas of redesignof the existing solution. The control list is constantly being revised as a result of theanalysis phase.

The iteration involves the redesign and implementation of a task from projectcontrol list, and the analysis of the current version of the system. The goal for the designand implementation of any iteration is to be simple, straightforward, and modular,supporting redesign at that stage or as a task added to the project control list. The codecan, in some cases, represent the major source of documentation of the system. The analysis of an iteration is based upon user feedback, and the program analysis facilitiesavailable. It involves analysis of the structure, modularity, usability, reliability, efficiency,and achievement of goals. The project control list is modified in light of the analysisresults.

During the implementation of the project by this approach, a step called V&V i.e.Verification and Validation is carried out at certain intervals.

Project Schedule :

Stage	Estimated Time
Requirement Gathering	25 Jan 2014 - 2 Feb, 2014
System Design	5 Feb, 2014 – 25 Feb, 2014
Coding	1 March, 2014 – 28 March, 2014
Testing & evaluation	1 April, 2014 – 10 April, 2014
Deployment	10 April, 2014 – 13 April, 2014

3.0 Overall Description

3.1 Product Perspective

A Recommender system for music data is proposed which assists customers in searching music data and provides result with items resulting in own user preference. This system first extracts unique properties of music like pitch, chord, and tempo from the music file using a CLAM annotator software tool.This extracted data is then stored on the database. Each stored property is analysed using content based filtering and interactive genetic algorithm.

3.2 Methodology

The following *Fig 3.2.1* describes the phases involved into the project. These phases are music feature extraction, selection phase, crossover phase and matching phase.

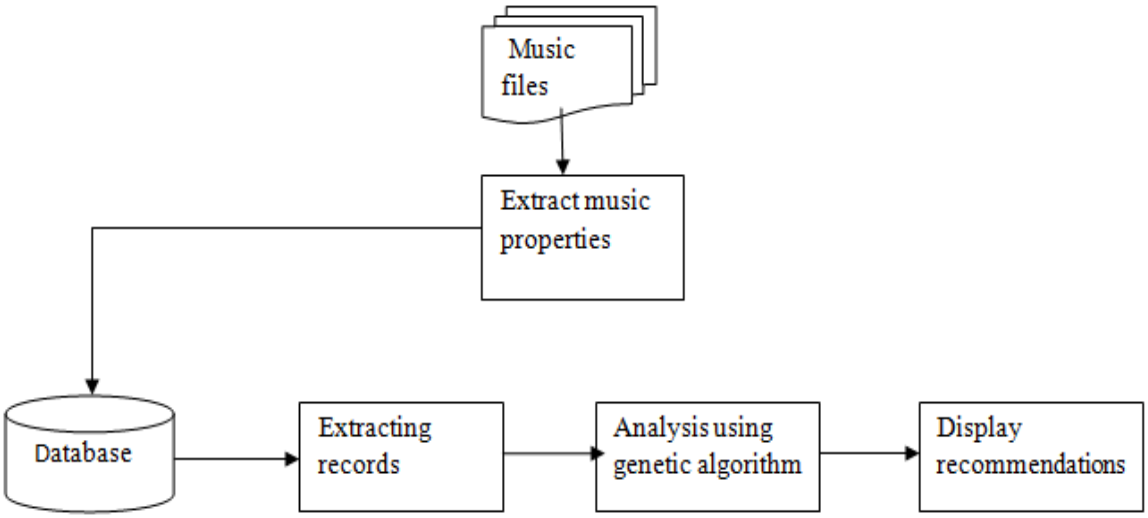


Fig 3.2.1: Phases involved in proposed work

1. **Music feature extraction:**

In this phase, system is going to extract unique properties of songs like tempo, chord, pitch etc using software tool called as 'CLAM Annotator' tool and we store these values in the database. Those values will act as an initial input for the next phases. [3]

2. **Selection phase:**

Music features are extracted using CLAM software. Truncation selection is used in this system, where a threshold is set for music files which are rated more than the threshold value is selected. Those records which fall below threshold value are not selected and are ignored[1],[2].

3. **Crossover phase:**

The BLX – alpha crossover algorithm is used since extracted features are real numbers. Hence crossover is performed with this algorithm resulting in new generation. The selected individual are given to the below BLX- alpha crossover algorithm. This algorithm is used to generate new off springs after the crossover step[1],[2].

4. **Matching phase:**

This phase finds the similar items stored in database to the newly generated music features. Once similarity is found those items are recommended to the user. This phase uses Euclidean distance between two offspring and distance between each feature of the two offspring is calculated, resulting value is used to match the records stored in the database. Those records are compared with the resulting value.

3.3<u>Requirement Specification</u>

HARDWARE SPECIFICATION:

1.Operating System: 32 or 64 bit Windows XP/7/8

2.Processor: All Intel or AMD 1 GHz

3.RAM: 256 MB

SOFTWARE SPECIFICATION:

1.CLAM Annotator Tool: required for music feature extraction

2.Database Management Tool: File System

3.4 Product Functions:

The project implements some functions in order to accomplish required tasks. These functions constitute a basis for the whole system. These functions can be stated as:

- **Submission of Preferences and Rating to songs by User**

  The software generates an interface which includes list of songs which the user can rate. Ratings are adjusted on the scale of 1 to 10.User can give ratings to single or multiple songs.

- **Recommendation of Similar Songs to User**

  The system retrieves the ratings given by the user and prefers the list of similar songs according to the genetic algorithm mentioned in the methodology.

3.5Constraints:

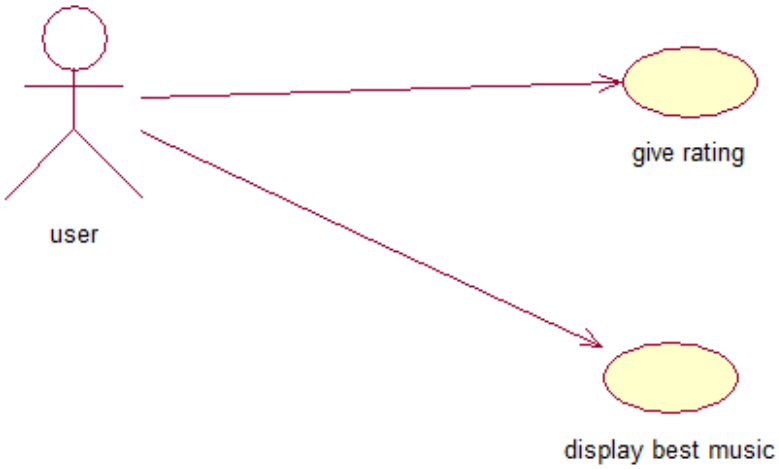
- **Hardware Limitations**

  There is no limitation in the operating system in which the project will work. Users can access the system without any internet browser.

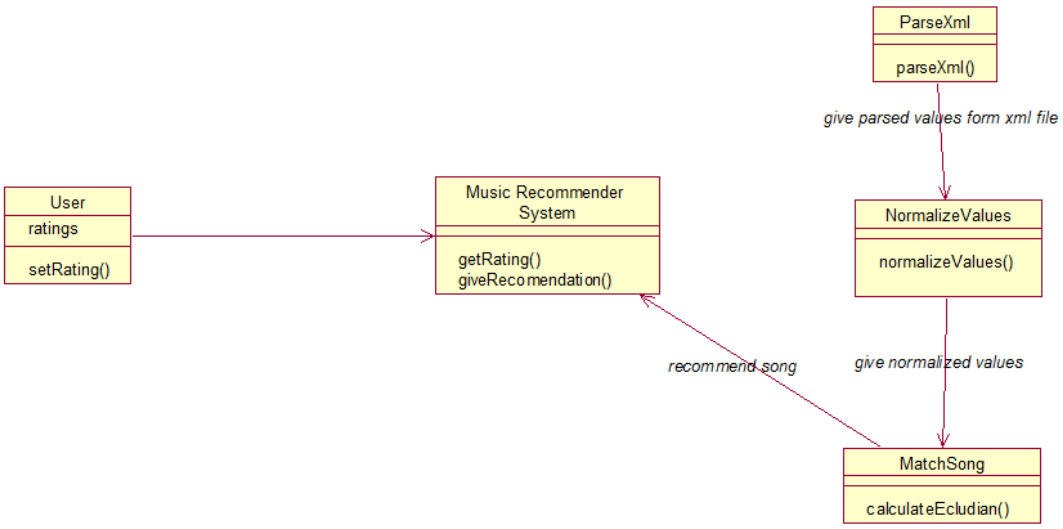
- **Authentication Policy**

  The software has no authentication policy for particular type of user.Any user can give the ratings and get back similar songs recommended by the system.

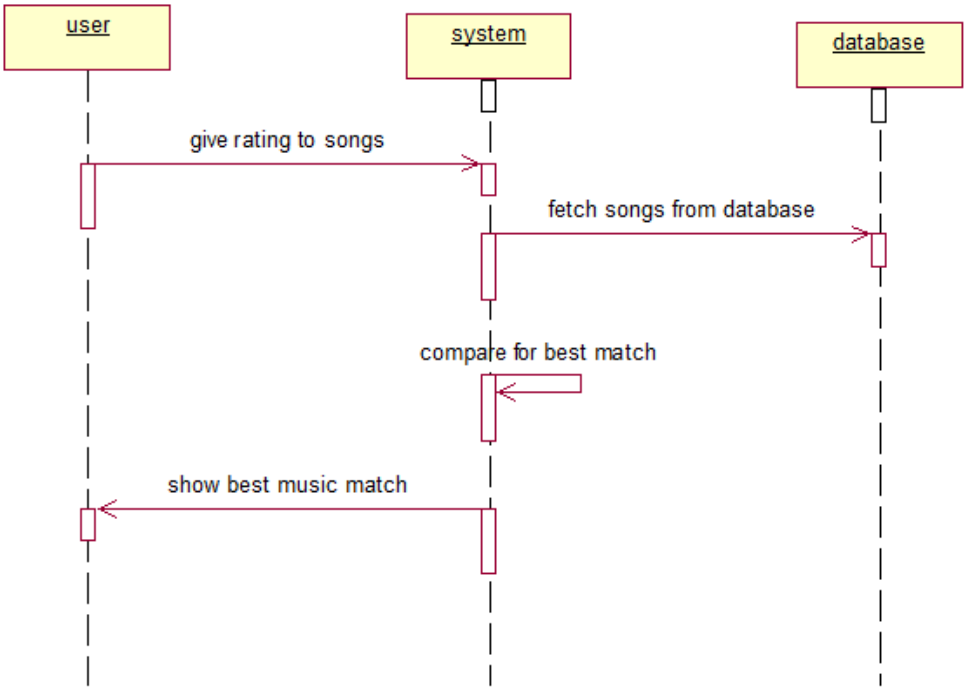
3.6Use Case Diagram:



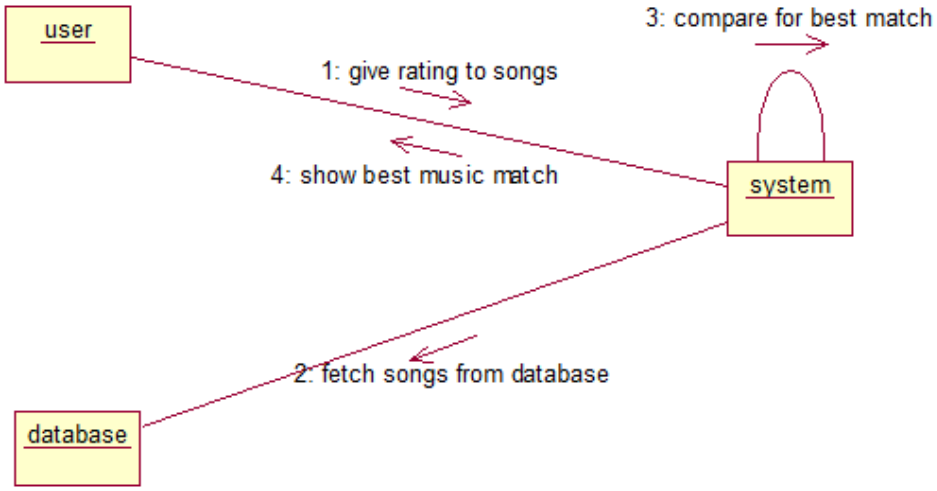
3.7 Class Diagram:



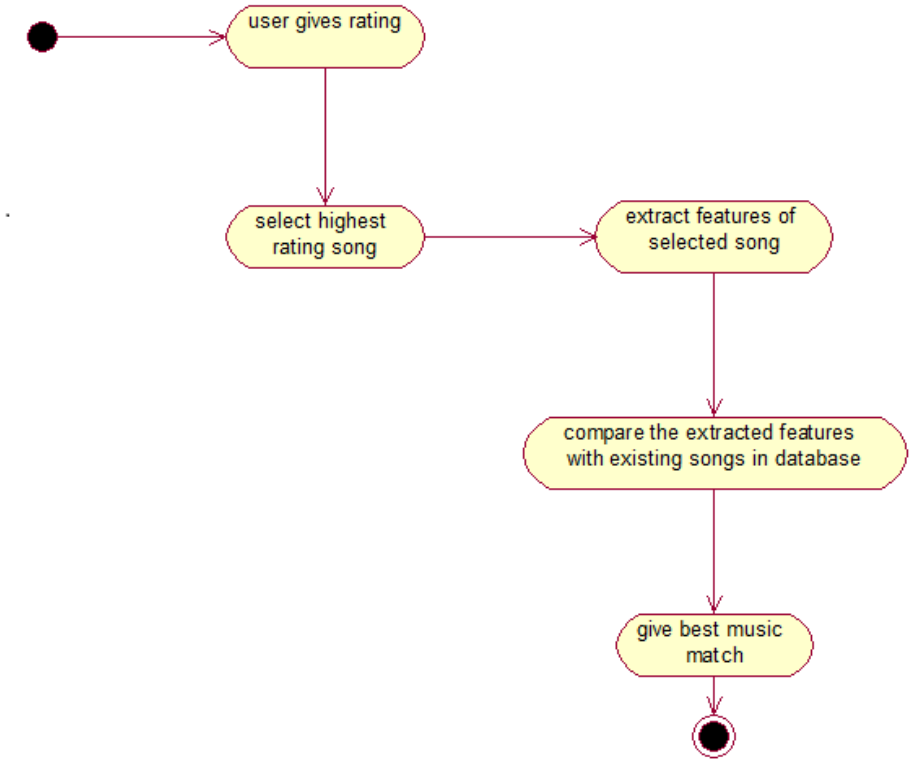
3.8 Sequence Diagram:



3.9Collaboration Diagram:



3.10Activity Diagram:



3.11 Source Code Documentation

In this implementation of we have used many built-in packages from Java and also created different functions.

1. **Packages from Java**

- java.io
- javax.xml.parser
- javax.swing
- java.org.w3c
- java.util

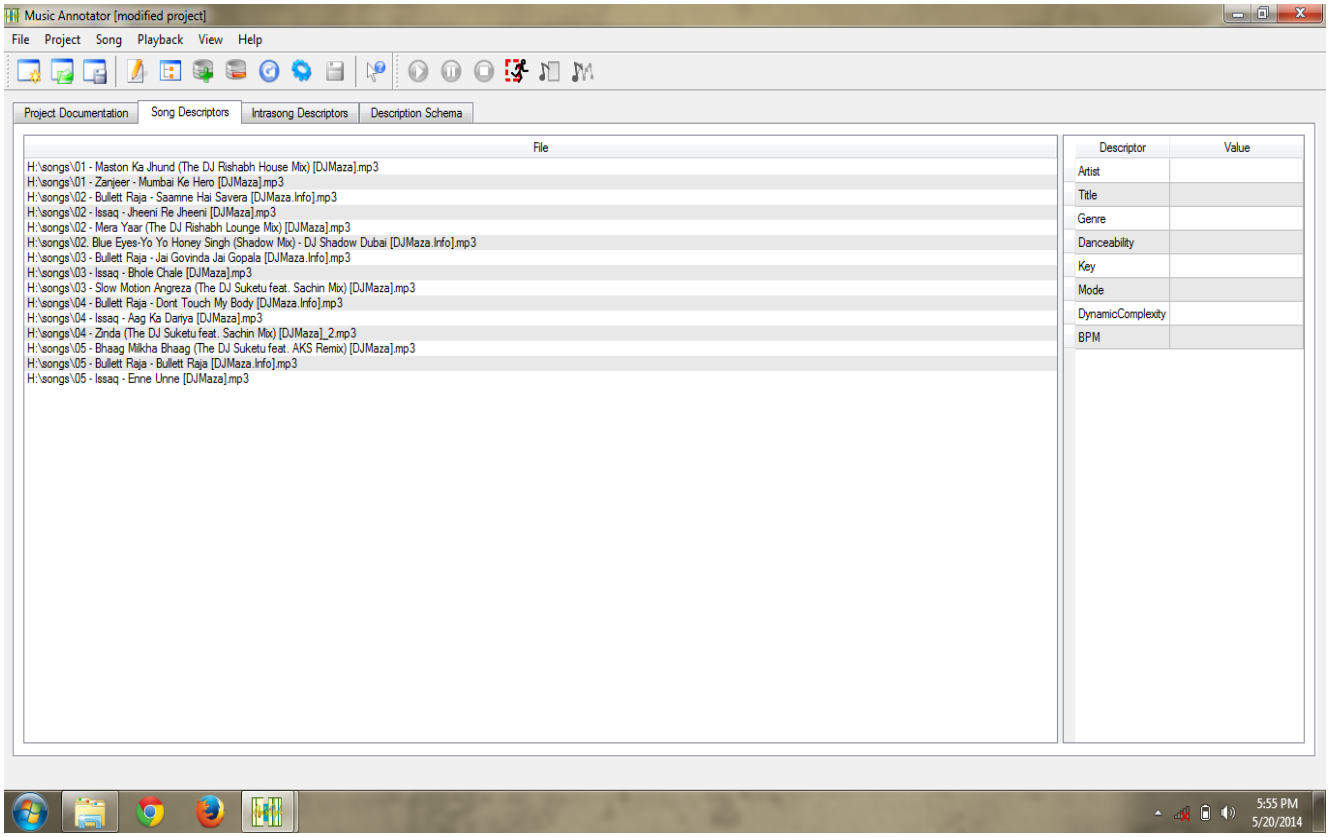
2. **Major Classes**

- ReadXML
- Normalise
- CompData
- Music

4.0SCREENSHOTS

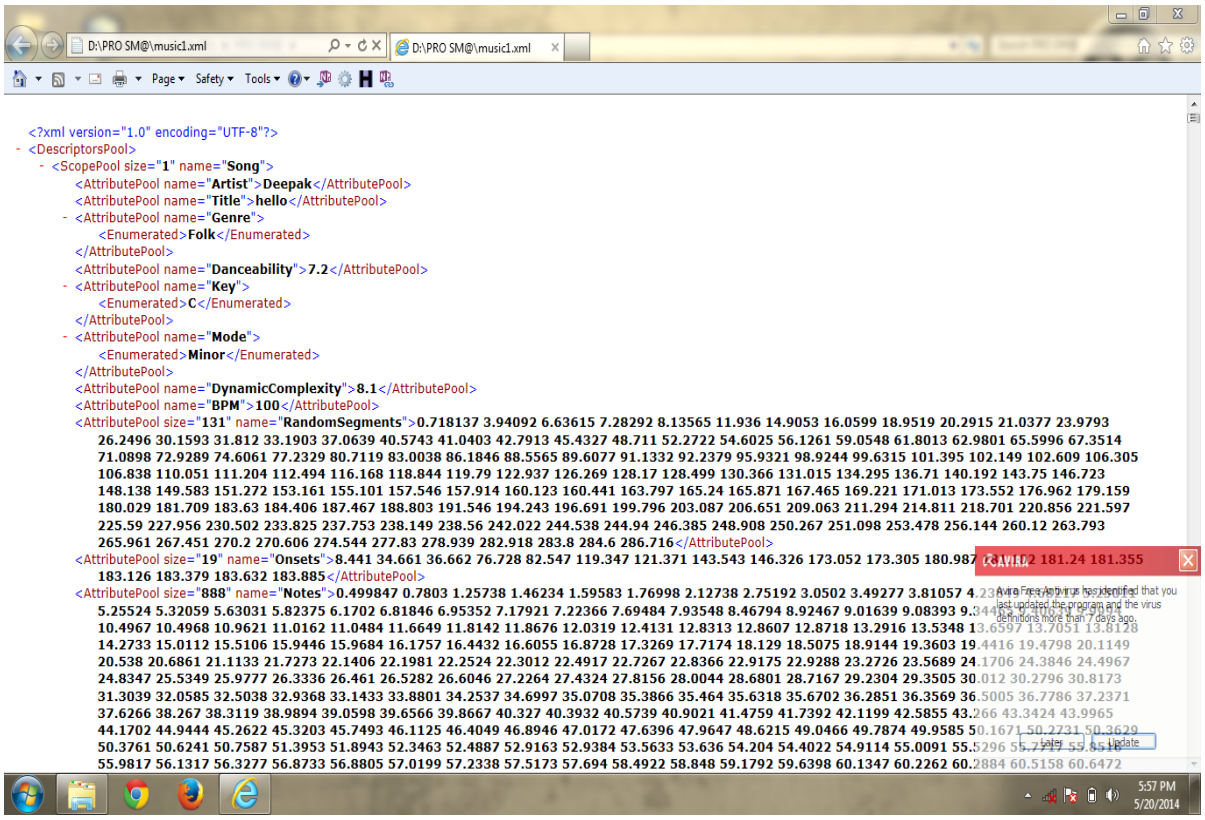
4.0.1:CLAM Music Annotator:

CLAM Music Annotator is a software tool which extracts the music features of the given songs and provide XML file containing music properties.



4.0.2:Music XML file:

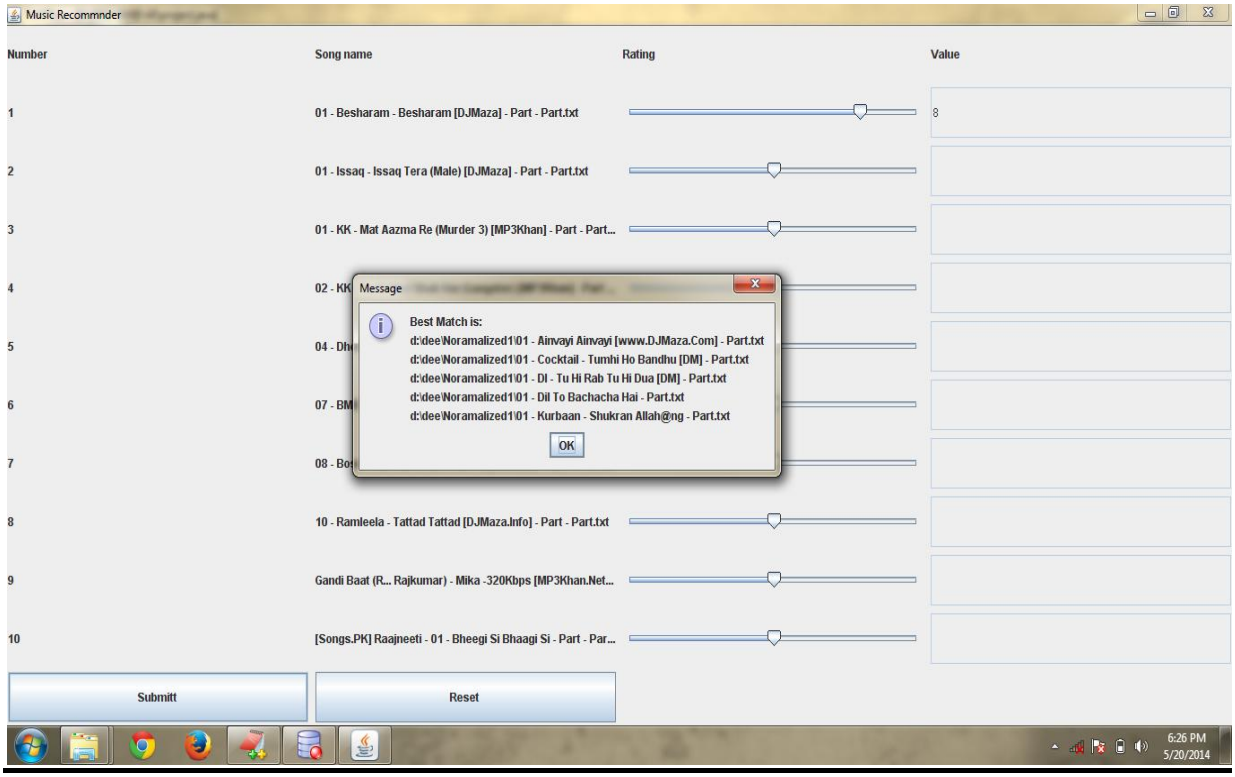
This is a XML file contains music features and is generated from CLAM Music Annotator.



```
<?xml version="1.0" encoding="UTF-8"?>
- <DescriptorsPool>
  - <ScopePool size="1" name="Song">
      <AttributePool name="Artist">Deepak</AttributePool>
      <AttributePool name="Title">hello</AttributePool>
    - <AttributePool name="Genre">
        <Enumerated>Folk</Enumerated>
      </AttributePool>
      <AttributePool name="Danceability">7.2</AttributePool>
    - <AttributePool name="Key">
        <Enumerated>C</Enumerated>
      </AttributePool>
    - <AttributePool name="Mode">
        <Enumerated>Minor</Enumerated>
      </AttributePool>
      <AttributePool name="DynamicComplexity">8.1</AttributePool>
      <AttributePool name="BPM">100</AttributePool>
      <AttributePool size="131" name="RandomSegments">0.718137 3.94092 6.63615 7.28292 8.13565 11.936 14.9053 16.0599 18.9519 20.2915 21.0377 23.9793
        26.2496 30.1593 31.812 33.1903 37.0639 40.5743 41.0403 42.7913 45.4327 48.711 52.2722 54.6025 56.1261 59.0548 61.8013 62.9801 65.5996 67.3514
        71.0898 72.9289 74.6061 77.2329 80.7119 83.0038 86.1846 88.5565 89.6077 91.1332 92.2379 95.9321 98.9244 99.6315 101.395 102.149 102.609 106.305
        106.838 110.051 111.204 112.494 116.168 118.844 119.79 122.937 126.269 128.17 128.499 130.366 131.015 134.295 136.71 140.192 143.75 146.723
        148.138 149.583 151.272 153.161 155.101 157.546 157.914 160.123 160.441 163.797 165.24 165.871 167.465 169.221 171.013 173.552 176.962 179.159
        180.029 181.709 183.63 184.406 187.467 188.803 191.546 194.243 196.691 199.796 203.087 206.651 209.063 211.294 214.811 218.701 220.856 221.597
        225.59 227.956 230.502 233.825 237.753 238.149 238.56 242.022 244.538 244.94 246.385 248.908 250.267 251.098 253.478 256.144 260.12 263.793
        265.961 267.451 270.2 270.606 274.544 277.83 278.939 282.918 283.8 284.6 286.716</AttributePool>
      <AttributePool size="19" name="Onsets">8.441 34.661 36.662 76.728 82.547 119.347 121.371 143.543 146.326 173.052 173.305 180.987 181.102 181.24 181.355
        183.126 183.379 183.632 183.885</AttributePool>
      <AttributePool size="888" name="Notes">0.499847 0.7803 1.25738 1.46234 1.59583 1.76998 2.12738 2.75192 3.0502 3.49277 3.81057 4.
        5.25524 5.32059 5.63031 5.82375 6.1702 6.81846 6.95352 7.17921 7.22366 7.69484 7.93548 8.46794 8.92467 9.01639 9.08393 9.
        10.4967 10.4968 10.9621 11.0162 11.0484 11.6649 11.8142 11.8676 12.0319 12.4131 12.8313 12.8607 12.8718 13.2916 13.5348 1
        14.2733 15.0112 15.5106 15.9446 15.9684 16.1757 16.4432 16.6055 16.8728 17.3269 17.7174 18.129 18.5075 18.9144 19.3603 19.
        20.538 20.6861 21.1133 21.7273 22.1406 22.1981 22.2524 22.3012 22.4917 22.7267 22.8366 22.9175 22.9288 23.2726 23.5689 24.
        24.8347 25.5349 25.9777 26.3336 26.461 26.5282 26.6046 27.2264 27.4324 27.8156 28.0044 28.6801 28.7167 29.2304 29.3505 30.
        31.3039 32.0585 32.5038 32.9368 33.1433 33.8801 34.2537 34.6997 35.0708 35.3866 35.464 35.6318 35.6702 36.2851 36.3569 36.
        37.6266 38.267 38.3119 38.9894 39.0598 39.6566 39.8667 40.327 40.3932 40.5739 40.9021 41.4759 41.7392 42.1199 42.5855 43.
        44.1702 44.9444 45.2622 45.3203 45.7493 46.1125 46.4049 46.8946 47.0172 47.6396 47.9647 48.6215 49.0466 49.7874 49.9585 5
        50.3761 50.6241 50.7587 51.3953 51.8943 52.3463 52.4887 52.9163 52.9384 53.5633 53.636 54.204 54.4022 54.9114 55.0091 55.
        55.9817 56.1317 56.3277 56.8733 56.8805 57.0199 57.2338 57.5173 57.694 58.4922 58.848 59.1792 59.6398 60.1347 60.2262 60.
```

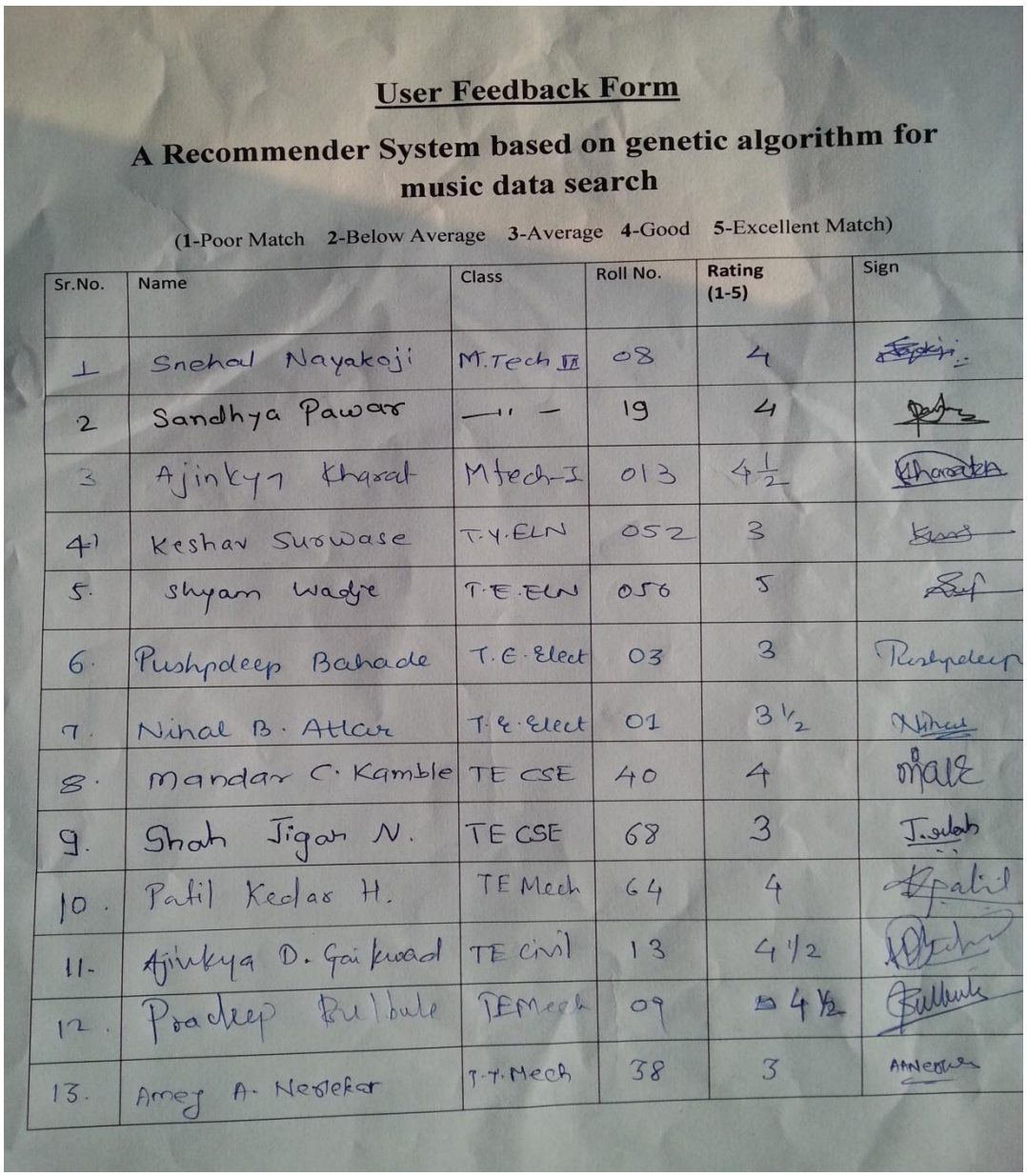
4.0.3:System Interface:

This is an interface to recommend songs to user.



4.0.4:Perfomance/Feedback

This is feedback given by random sample of users for above proposed system.



User Feedback Form

A Recommender System based on genetic algorithm for music data search

(1-Poor Match 2-Below Average 3-Average 4-Good 5-Excellent Match)

Sr.No.	Name	Class	Roll No.	Rating (1-5)	Sign
1	Snehal Nayakoji	M.Tech II	08	4	
2	Sandhya Pawar	—"—	19	4	
3	Ajinkya Kharat	Mtech-I	013	4½	
4)	Keshav Surwase	T.Y.ELN	052	3	
5.	shyam Wadje	T.E.ELN	056	5	
6.	Pushpdeep Bahadde	T.E.Elect	03	3	
7.	Nihal B. Attar	T.E.Elect	01	3½	
8.	Mandar C. Kamble	TE CSE	40	4	
9.	Shah Jigar N.	TE CSE	68	3	
10.	Patil Kedar H.	TE Mech	64	4	
11.	Ajinkya D. Gaikwad	TE civil	13	4½	
12.	Pradeep Bulbule	TE Mech	09	4½	
13.	Amey A. Nerlekar	T.Y.Mech	38	3	

5.0 Conclusion and Future Scope

The software represents a novel recommender system for music data. Our proposed system is able to recognize the preferences of user and then intuitively recommend music tracks appropriate for their present atmosphere.

To construct the system, we incorporated the main interactive genetic algorithm-based engine with the content based filtering method. First, the music features of each music track are extracted. We then apply the interactive genetic algorithm to obtain the most appropriate tracks to recommend to users.

Based upon the ratings of random sample of users, it can be concluded that the software is effective.

The software can be extended to a web application where user can submit his/her own favourite song to the system and get recommendations. For processing of large data, we can extend our project to parallel system for fast processing.

References

[1] Hyun-Tae Kim,EungyeongKim,Jong-Hyun Lee,ChangWookAhn "A Recommender System based on Genetic algorithm for music data.",*2nd internation conference on computer engineering and technology*,vol.6 no 415 2010.

[2]Michael Negnevitsky, "*Artificial Intelligence-A guide to intelligent Systems*" Pearson Education Limited,England2002.

[3] httpreference:http://www.clam-project.org.