A Project For

Antakshari Assistant

Indian Institute of Technology (ISM) Dhanbad

Guided by
Prof. Chiranjeev Kumar

Department
Department of Computer Science and Engineering , IIT (ISM) Dhanbad

Submit by:
Krishan Kant Mehra (19JE0450)
Ankit Kumar (19JE0141)

ACKNOWLEDGEMENT

We would like to express our special thanks to our guide Prof. Chiranjeev Kumar for his invaluable support, encouragement of working harder throughout the journey of the project.

We specially acknowledge him for his advice, supervision and the vital contribution as and when required during the project. His involvement with originality has triggered and nourished our intellectual maturity that will help us for a long time to come. We are proud to record that we had the opportunity to work with an exceptionally experienced Professor like him.

Signature

(Prof. Chiranjeev Kumar)

TABLE OF CONTENT

Motivation

In this era of internet and social media, we have forgotten traditional games like antakshari. Antakshari can be seen as a game with lots of interactions. This game is connected with our traditions.

Our way of addressing this problem is to combine today's technology to rejuvenate the forgotten game.One way of achieving this goal is to build a platform to assist throughout antakshari game. This would make the game more flexible for players as players don't have to be in the same geographical location to play antakshari.

- Our main aim is to apply the concepts of machine learning and software engineering to make a reliable antakshari assistant platform.

Introduction

The software product to be produced is an Antakshari Assistant which would let players play Antakshari and it will recommend songs to them. The system would have a web based interface and for recommendation sub-part, it would use Machine Learning.

Major sub-parts of antakshari assistant:

1. Database

   This sub-part will store and provide song details like song name,its lyrics, genres etc. to the interface and recommendation subpart. It would store user/Player details like user name, his / her song preferences etc.

2. Music Recommendation system

   This sub-part would be majorly dependent on machine learning. It would use a database to keep track of the user's preference. By using a music recommender system, the music provider can predict and then offer the appropriate songs to their users based on their preference. The main criteria for recommendation would be the genre of song the user has listened to previously.

3. Audio-to-text convertor:

   This sub-part would convert the user's microphone input audio into text form and provide this data to other subparts.

4. Interface for antakshari

   This part would provide a web application Interface for the user. Users would only be able to interact with the system through this interface.

   Through this user would be able to login or sign up into the system and create or join an antakshari room. It would keep track of the user's song lyrics as they are playing Antakshari and by using a recommendation system and lyrics it would provide next song details to the user.

5. Game Manager

   The game manager system would control the antakshari game features like player terms , timer, lose / win conditions etc.

Overview Of Antakshari Assistant Software

There is only one entity : the user.

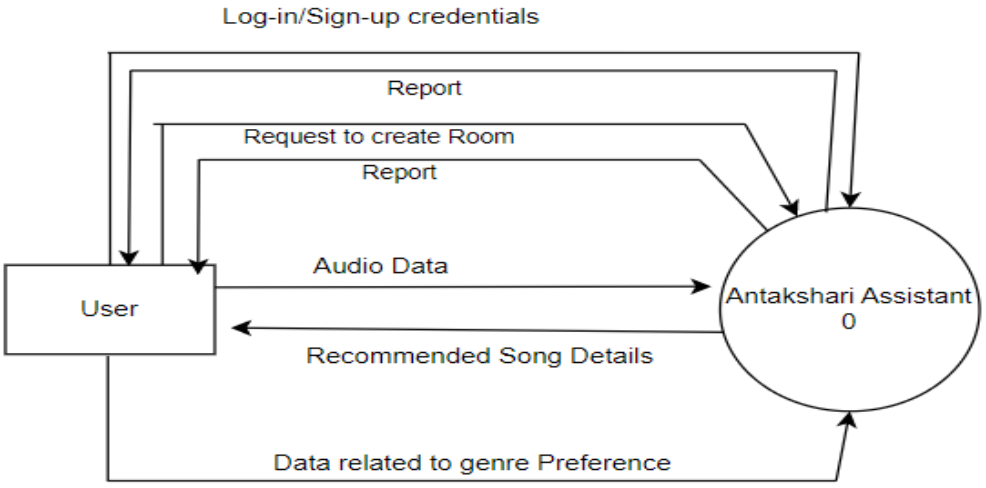
For login / signup, the user can send his / her credential details through login/sign-up page to the system.System would communicate with the database to confirm user credential. If the user is authenticated then the system would redirect the user to the homepage of the system.

For playing antakshari users have to request the system to create a room for playing. Then the system would create the room.Then the user has to send the invitation to other players.Players can join in the game room using an invitation from the user who created the room.

After each player has joined the game room, the game can be started. The Game Manager Subsystem will control all features of this game. During a player's term, the system will send data to the recommendation system. In response, the recommendation system would send back details of recommended songs.

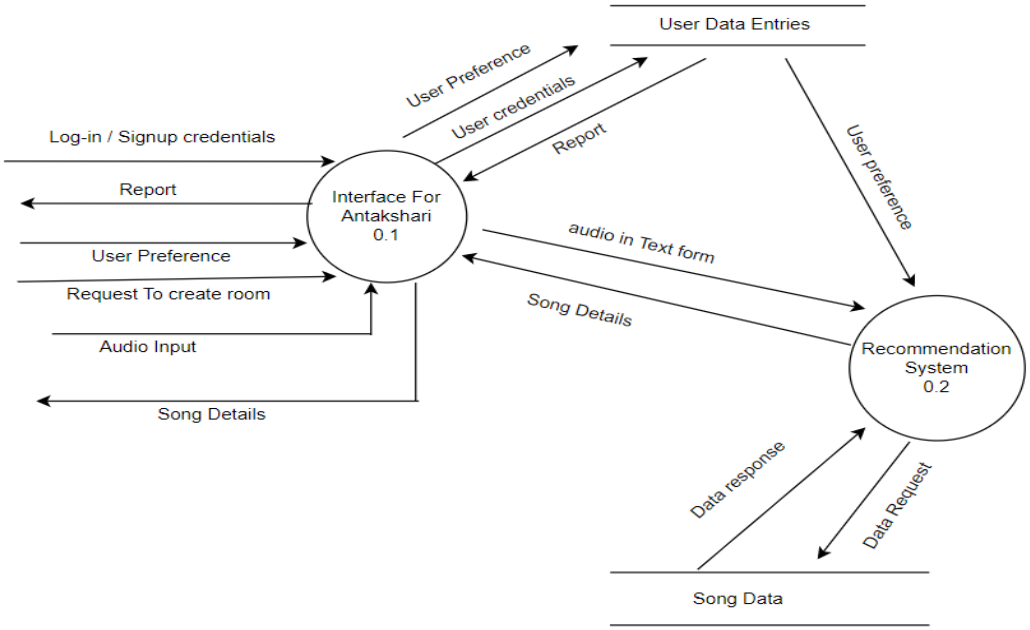
The whole system process can be understand by Given DFD (Data Flow Diagram) of System :

Level 0 DFD (Context Diagram)



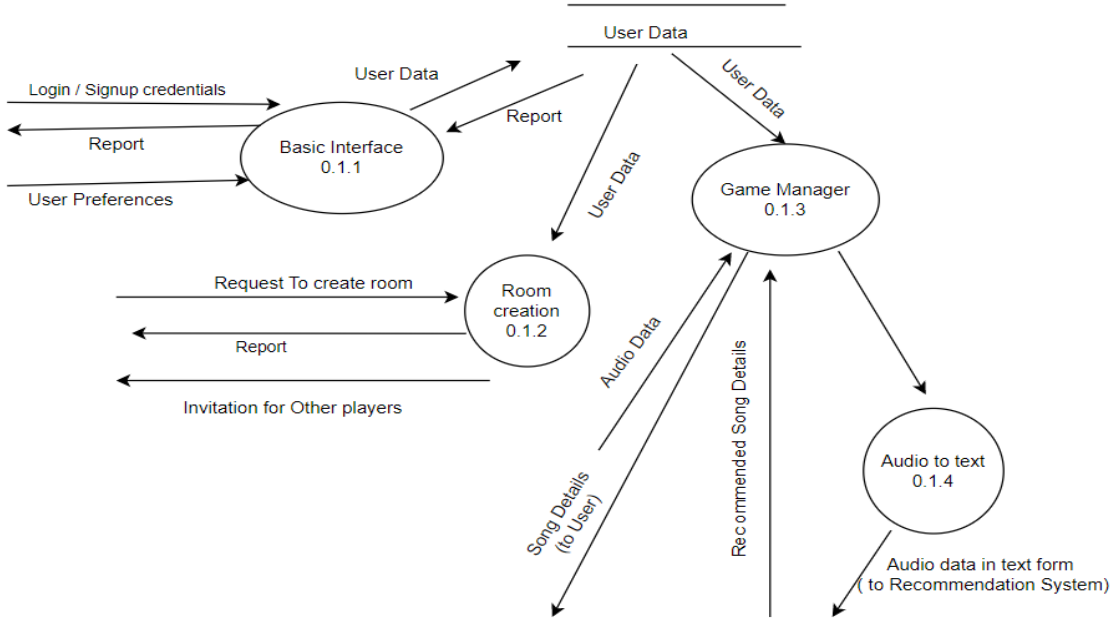
Level 0 Diagram

Level 1 DFD :



Level 1 DFD

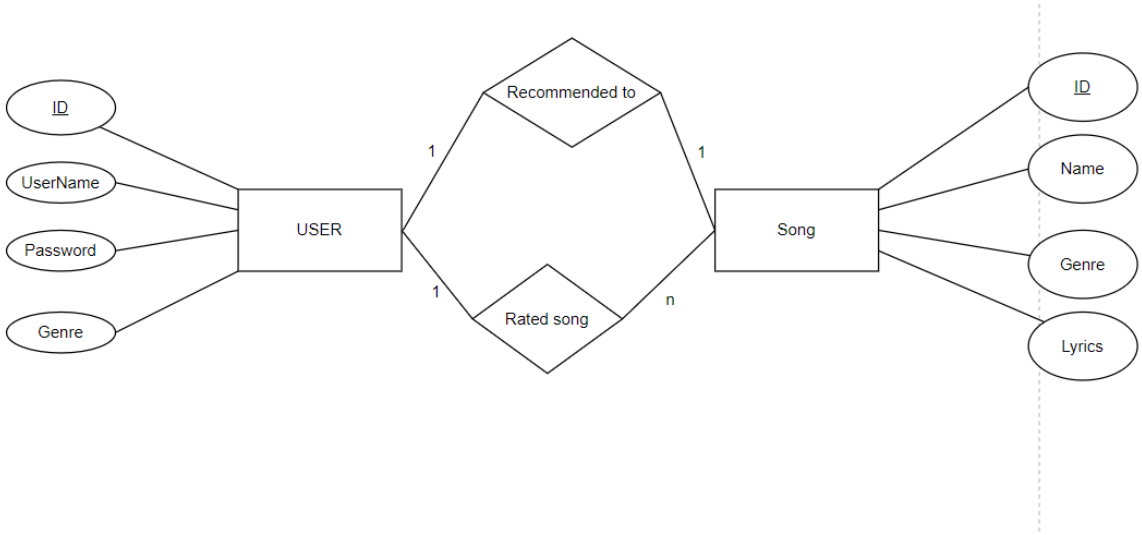
Level 2 DFD :



LEVEL 2 DFD

Database structure

ER diagram (Entity - Relation Diagram):



Data Description

User Table

Data Item	Type	Description
ID	Integer	Unique for every user
Username	Text	Name of User
Password	Text	User password in Encrypted form
Genres	Text	Preferenced genre by user

Song Table

Data Item	Type	Description
ID	Integer	Unique for every Song
Name	Text	Name of Song
Genre	Text	Genre of song
Lyrics	Text	Song Lyrics

User - Song relation Table

Data Item	Type	Description
User ID	Integer	Forgien key for user ID
Song Id	Integer	Forgien key for Song ID
Rating	Integer	Rating of song by user

Description of Recommendation System

For recommending songs, the Game manager sub-part would send a request to the recommendation sub system through API. With this request the game manager would send the data like last words, player preference etc. By using this data recommendation system would recommend the song and would send back the recommended songs details to the game manager.

Working:

With request Game Manager would send the following data to recommendation System:

1. Last words of Previous player song
2. Current Player Details

System would get the song details from the database.

After that, the recommendation system would filter the song on the follwing basis:

1. Song should contain any of Last words of Previous player song
2. Song genres should match with the current player's preference genre.

This process would return to the system a list of songs which satisfies the above criteria.

The system would use item-based collaborative KNN filtering algorithm for recommending the song from the list of songs.

Advantages of KNN Algorithms:

1. No Training Period- KNN modeling does not include training period as the data itself is a model which will be the reference for future prediction and because of this it is very time efficient in terms of improvising for a random modeling on the available data.
2. Easy Implementation- KNN is very easy to implement as the only thing to be calculated is the distance between different points on the basis of data

of different features and this distance can easily be calculated using distance formula such as- Euclidean or Manhattan.

Working of KNN algorithm

Step 1:

Get the data from the user-song rating table and create a user-item matrix with item as row and user as column.

For all NAN values put zero.

Each row is representing a vector for a song.

Step 2:

Find similarity between songs on the basis of center cosine similarity.

Step 3:

For all songs find rating prediction using KNN algorithm.

To calculate prediction rating, for each songs which user has not rated do the following steps:

- Get the songs list which the user has rated.
- Get center cosine similarity between them and song for which prediction rating has to be calculated.
- Prediction rating of the song would be calculated by blow formula:

$$\hat{r}_{ui} = \frac{1}{\sum_{j \in \mathcal{S}(i)} s_{ij}} \sum_{j \in \mathcal{S}(i)} s_{ij} r_{uj}$$

Where

r_{ui} = Predicted rating by user u for song i

S_{ij} = Similarity between song i and j

R_{uj} = Rating by user u for song j

Step 4:

Sort on the basis of prediction rating and return K songs with maximum rating.

From this process the system would generate a list of recommended songs and send their details back to the Game Manager subsystem.

Abbreviations

DFD	Data Flow Diagram
ER Diagram	Entity Relation Diagram
API	Application programming interface
KNN	K-nearest neighbor

Future Work

1. **Design the System Interface (3 Week)**

   We have to design the web interface for users to interact with the system.

2. **Set up Database (1 weeks)**

   We have to Set up the Database to store users , songs and rating details.

3. **Design APIs (4 weeks)**

   We have to design APIs so that these subsystems can interact with each other.

4. **Testing (2 weeks)**

   Test the full-scale system at the end.

References

- https://www.youtube.com/watch?v=h9gpufJFF-0