**CO303 DATABASE MANAGEMENT SYSTEMS LAB**

**A REPORT ON THE PROJECT ENTITLED**

**“Grey Vibrant”**



**SUBMITTED BY**

**NAME1 – Shrinidhi Anil Varna**

**NAME2 - Sayan Biswas**

**V SEMESTER B-TECH**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA, SURATHKAL**

**2019-2020**

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **CONTENT** | | **PAGE NUMBER** | |
| 1 | Abstract | | 3 | |
| 2 | Introduction | | 4 | |
| 3 | Database Design | | 6 | |
| 3.A | ER Diagram | | 6 |
| 3.B | Relational Schema | | 7 |
| 3.C | Normalized tables | | 8 |
| 4 | Modules | | 10 | |
| 5 | Tools | |  | |
| 6 | Results | |  | |

**1. ABSTRACT**

Grey Vibrant is a music streaming android application.

There are two prototypes User and Artist. Artists can upload their own songs and manage them. Users can create their own playlist, follow desired artists and listen to songs of their choice live by adding them to a queue.

**2. INTRODUCTION**

Describe your idea here.

Describe what is the use of your project in 1- 2 pages.

Grey Vibrant is a music streaming android application that lets you play songs that you desire. It lets you follow artists of your choice and recommends you the songs of the followed artists. The remaining songs which come under unfollowed artists are also shown and can always be listened to. As an artist you can manage your album by adding a song, along with its description and view them in your personal album.

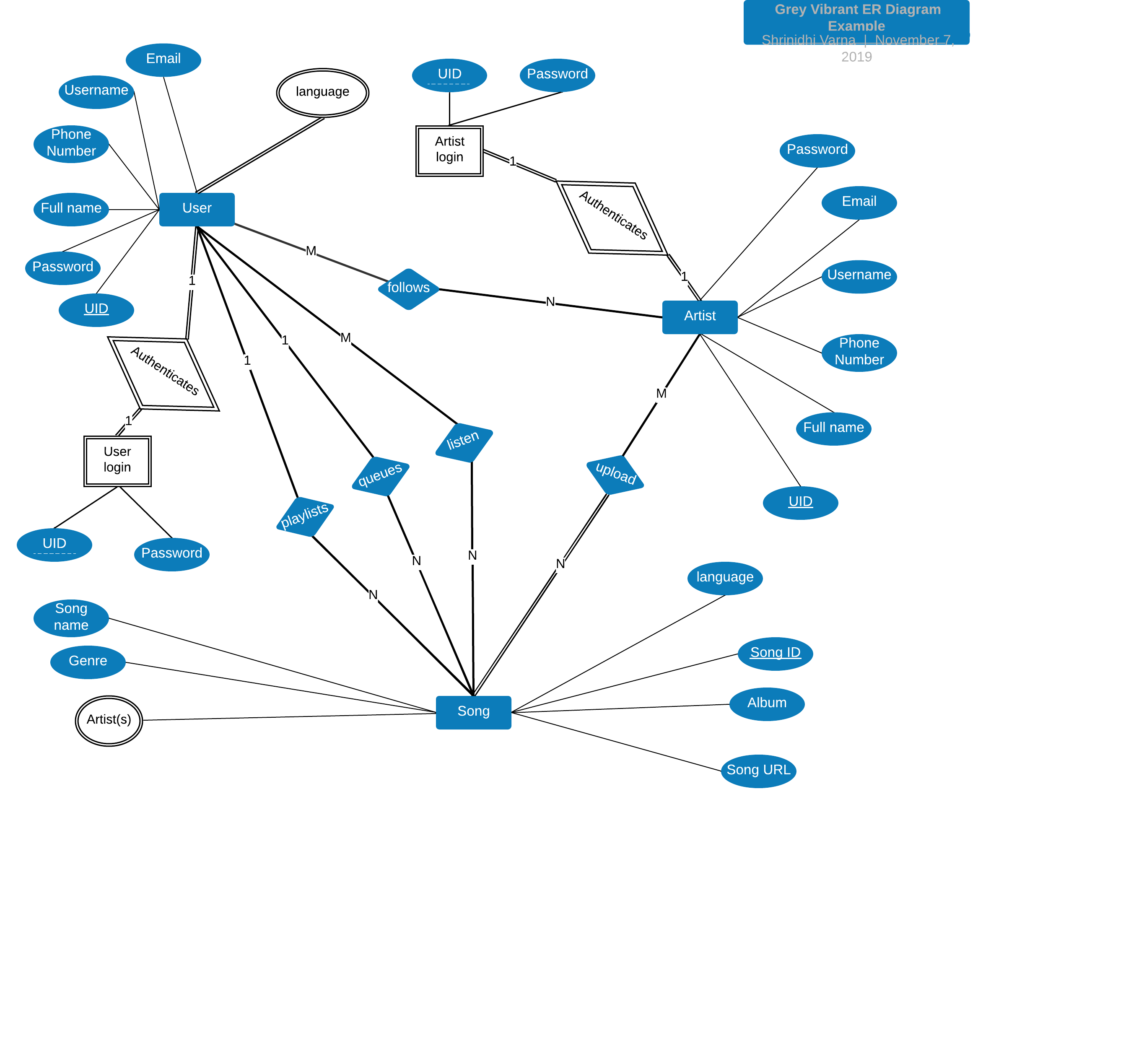
For a user, some special features that are provided include, adding a song to a personal playlist, adding it to a queue to listen later, and maintaining a history of the songs that you have heard before.

Apart from this a user can always unfollow an artist, delete a song from a playlist, queue or from song history. An artist can also delete a song from his/her personal album in case it was added by mistake or due to some other reason. A search functionality is also implemented to search a song on the basis of artist name, album, song name, language and genre.

**3. DATABASE DESIGN**

**3.A ER DIAGRAM**

Paste your ER Diagram here.



**3.B RELATIONAL SCHEMA**

Describe about all the tables.

*1. user\_registration (UID, fullname, phNo, username, email, password, playlist\_name)*

Maintains a record of all the users after their registration.

2. *user\_language (UID, language)*

Maintains a record of all the languages that a user wants to listen. (song language)

3. *user\_login (UID, username, password, foreign* *key (UID) references* *user\_registration(UID))*

A table that keeps the information required to authenticate a user.

4. artist\_registration(AID, artistname,email, password, phNo, fullname)

Maintains a record of all the artist after their registration.

5. *artist\_login (AID, artistname, password, foreign* *key (UID) references* *artist\_registration(AID))*

A table that keeps the information required to authenticate an artist.

6. *song (SID, songname, songurl, genre, language, album)*

Maintains a record of all the songs that have been uploaded by various artists registered in the application.

7. *song\_artists (SID, AID)*

A table that links a song and its corresponding artist.

8. *follow\_artists (UID, AID)*

A table that provides a link between a user and the artist(s) he/she follows

9. *listens (UID, SID)*

A table that maintains all the song(s) that a user has already listened to.

10. *playlist (UID, SID)*

A table that maintains all the songs that have been added to the playlist by the user.

11. *queue (UID, SID)*  
Contains all the songs that are currently queued for streaming for the user by the music player.

**3.C NORMALIZED TABLES**

Describe about normalization briefly.

Describe your 3NF tables here.

Database normalizationis a database schema design technique, by which an existing schema is modified to minimize redundancy and dependency of data. Normalizationsplit a large table into smaller tables and define relationships between them to increases the clarity in organizing data. It is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.

**3NF tables of the database schema mentioned earlier**:

**1. user\_registration**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UID | fullname | phNo | username | email | password | playlist\_name |

**2. user\_language**

|  |  |
| --- | --- |
| UID | language |

**3. user\_login**

|  |  |  |
| --- | --- | --- |
| UID | username | password |

**4. artist\_registration**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AID | artistname | email | password | phNo | fullname |

**5. artist\_login**

|  |  |  |
| --- | --- | --- |
| AID | artistname | password |

**6. song**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SID | songname | songurl | genre | language | album |

**7. song\_artists**

|  |  |
| --- | --- |
| SID | AID |

**8. follow\_artists**

|  |  |
| --- | --- |
| UID | AID |

**9. listens**

|  |  |
| --- | --- |
| SID | UID |

**10. playlist**

|  |  |
| --- | --- |
| UID | SID |

**11. queue**

|  |  |
| --- | --- |
| UID | SID |

**4. MODULES**

Describe all the modules present in your project with screenshot.

The different modules present in the project are:

1. Upload: In this module, an artist can upload a song (mp3 file) along with its language, genre and songname to the database. A song is uploaded to a storage bucket and its URL is stored in the **song** table of the database. This URL is used for streaming music whenever a user listens to the song.

2. Music Stream: This module contains a music player that can play a list of songs back to back by streaming it live from the song URLs stored in the database. The functionalities available in this module are: play next, play previous, start, pause, pick a song from the queue for playing. This module also has a queue of songs shown to the user which were selected for streaming. These songs can be picked at random and played, played sequentially, deleted from the queue, more information can be provided about the songs in queue.

3. Home page: This page has a list of artists a user follows, unfollows, recommended songs, songs that can be still listened along with a search functionality provided to filter the results. The search works on the basis of artistname, songname, genre, and language. The artists who are followed can be unfollowed and the unfollowed artists can be followed just by clicking on their named cards. The songs that are displayed can be held pressed for a longer time to use few more functionalities of the module. They are: More information about the song, add song to playlist, add song to queue for playing.

4. Album: This module is provided to an artist to view all his uploads in his album. Search functionality is provided to filter the results as per artist’s choice. There is more information provided for each song and the song can be deleted from the album if the artist wants to.

5. Playlist: A playlist contains a list of songs that have been chosen by the user from a list of all songs available in the application. This playlist contains those songs which the user feels little special about. These songs can all fit in at one place called playlist. The functionalities provided in this module apart from displaying the playlist are: adding a song to the queue, deleting a song from the playlist, more information about the song and search functionality. The songs can be searched on the basis of song name, artist name, genre, and language.

6. Song history: This module contains all the songs that have been listened by the user. The functionalities provided here are adding a song to the queue, getting more information about a song, deleting a song from the history and searching songs based on songname, artistname, genre or language.

7. Profile page: This module is provided for both user and artist. The user and artist can see some personal information provided to the application at the time of registration. The user can additionally see the number of songs present in his/her playlist, history and number of artists followed. The artist on the other hand can additionally see the number of users who follow him/her, and number of uploads in the album.

**5. TOOLS**

**FRONT END TOOLS**

Describe the front end tools and framework.

**BACKEND TOOLS**

Describe about MySQL

**6. RESULTS**

Give a brief conclusion.

Paste screenshots of all the results you achieved.