IT 214 DBMS

Lab 7(a)

Prepared by: Group S6_T9

ID	Name
201901076	Utsav Ladani
201901090	Pandar Mayur
201901131	Bhavya Solanki
201901304	Dev Joshi

Dhirubhai Ambani Institute of Information and Communication Technology

26 Oct, 2021

1. Final Problem Description

The Online Music Management System will provide a large collection of songs to every user who is connected with our service over the internet. Users can search music and/or select from a given list of songs and that song will play in the appropriate app or system (web / android / desktop). As far as the IT214 subject is concerned, this project will be database oriented. Our goal is to make efficient database design, so that maximum users can connect with our service using minimum resources.

The system consist of three parts;

1. **User interface:** installed on device or accessed via browser
2. **Server:** connect application and database, handle user request, stream song, register user,etc.
3. **Database:** Store the songs and user information, provide some basic functions to modify and search into the database. It uses a PostgreSQL database.

Basic Workflows of Client Application

- **Register/SignUp:** Every user needs to register in application when he/she operates for the first time.Without registering users can listen to only 20 songs.

- **SignIn:** Every time when a user opens an application, he/she must have to sign in using their credentials.

- **Suggestion/Song List:** After signing in, users get the list of some song according to their preference and history.

- **Search Bar:** Search the song in our database using various filters like artist name, rating, views, type, date, movies, etc.

- **Music Player:** Play, pause, prev, next, queued, etc. functionalities are handled by the music player.

- **User Profile:** show the information about user, account type(premium/comman), update profile, etc.

- **Upload Song:** Users can upload songs on the server.

- **Money:** Give the money to artists according to views and rating.

Basic Workflows of Server

- **Authentication Handling:** all authentication related functionality like register user, check whether the user's credential is right or not, etc. are done by this part.

- **Search request:** Send the result of the search query to the application in JSON format.

- **Song request:** Start the streaming of songs on a particular user's device, download, upload can be done via this service.

- **Update:** Update song rating, views,etc. and user data also.

- **Money:** All money transactions controlled by this part.

Basic Workflow of Database

- **User:** Add, delete, update user and also update account to premium account.

- **Song:** Add, delete, update song.

- **Money:** Money transaction.

Popular Issues

- **UI/UX:** Modern apps must have to provide a customization of theme and layout because bright light can harm eyes at night and UI with dark color can not be visible in day. So instance switching of themes was needed. Different layouts give the diversity of system usage.

- **Suggestion:** Users don't want the same type of suggestion all the time. They want to customize the suggestion based on their own requirements. No any existing system have this kind of feature to customize the suggestion for each user according to user requirements

- **Filter:** Only Suggestions are not enough, people must need some filter to get a particular song list. All music platforms are providing different filters, but it's not sufficient. Users want to filter the songs according to the group of attributes. So I need some sort of feature to apply more than one filter on any list of songs, even in a playlist, albums ,search result, etc.

- **Offline Availability:** Copyright concern prohibits the user to download songs. Many places like gym, car on highway, train, flight etc. have low connectivity due to closed infrastructure or mobility of devices. In such cases users want the offline availability of songs. There is some smart system required so that users can download songs without violating copyrights.

- **Other:** Playlists are too important to feature in any music system. System must take care of this feature. Users should be able to share songs with some privileges. Users also want to gather online and organize a virtual party (party means one can play a song and other listen to it), podcast, etc. So the system adds these features so that users get a better experience.

Online Music Management System(OMMS) will help to categorize using song name, artist, albums, rating, type, time, type, number of listener, etc. Admin will be able to add songs. Users will be able to create their favourite playlists, Download and Share songs. This system also has features for singers and authors of albums. If some other singer wants to use a song or it's lyrics in his song, He will have to take permission from the owner of that song. This system will alert the owner of a song or album when someone publishes a song or album containing the song of the owner. Based on that if he has not taken permission from the owner he will be charged a penalty. We are providing **Repeat Current song, Repeat playlist, Shuffle playlist** like features for users. Users can make their own playlist by adding their favourite songs. The system also has a **play in background** feature so a user doesn't have to keep the application open while listening to music. To use this play in background feature the user has to buy premium.To set any song for caller tune(song for incoming call) the user has to buy premium.

There are several other applications in the market which are similar to our OMMS software.these applications/softwares provide several functionalities like search music,add different songs in the favourite song list,set caller tune,lyrics,etc.

Search Music: By this function you can search the music song whichever you want.

Favourite song list:some applications provide a Favourite song library/folder and in this library you can add the song so it notes the song to your favourite song.

Caller tune:this function provides the ability to set the song for incoming calls.

Lyrics:some other applications provide lyrics of songs.

To achieve this workflow in our system, We will need to design a database which includes several entities having relations internally. For users, we will need tables storing information about the user, about the premium accounts, about user credentials, past transection of users etc. For the song, we will need tables storing various information about the song like song type (pop, rock, rap, bhajan etc.), song language, album name, singer, whether the song is for premium users or not, artist name, premium price, views, likes, dislikes, rating etc. for the singers and artists we will need tables storing their song names, their albums, their details, production company name etc. For the production company we will need tables storing information about songs or albums they have released, artist or singer names of that song/album, price of that song/album etc. For the admin of the system we will need several functions for adding songs in system, deleting songs in system, updating details of songs etc. We will need transaction management in our system.

For the users, We will need different sets which store information about the user. User table will have attributes like User ID, User Name, Email ID, Mobile number, etc. We will also need to define a set which tells which users have premium accounts and which have not. It will have attributes like User ID, User Name, Premium Details, Premium time etc. We will also have to store the privileges (like view add, delete, update, edit etc.) for different users(not listener only) of the system. User and Premium user table will define a relationship named Premium user account.

For the Artists, We will need different sets which store information about the artists. We will need a table which will store information about the artists which will have attributes like Artist ID, Artist Name, Produced songs etc. We will define a set named album which will store albums present in the system which will have attributes like album name, artist name, artist ID, album ID, Count of songs present in a album etc. We will need a set which will store the rating of the album which will have attributes like Album ID, Album rating etc. We will define a relationship between artist and album which will give us a relation named Artist’s Album.

We will define several sets to store the information about the production company. We will define a set named production company which stores information about the company. It will have attributes like company name, company ID, produced albums, employers(artists). We will define a relation between Artists and the production companies which will give us a relationship named Job details. We will also need to store information about the profit which each produced album made for the production company. It will have attributes like company name, company ID, album name, album ID, Profit etc.

Music is the main component of our system. We should handle music related things properly. To store the Music, we will need a set named Music. It will have attributes like Music name, Music ID, Artist name, Artist ID, company name, Company ID etc. To store different languages of the music we will need a set named Music language.It will have attributes like Music ID, Music name, Music language etc. To store the type of the music (rap, rock, pop, bhajan etc.) we will need a set named Music Type. It will have attributes like Music ID, Music name, Music Type etc. All these sets should be connected to each other for the query optimization because the listener of our system will search music based on various things. To store whether a song is for premium users or not, we will create a set named premium songs which will have attributes like Music ID, Music Name. To store views and likes of the songs we will need a set which will store likes and views of the songs. It will have attributes like Music ID, Music name, Views, Likes, Ratings etc.

Users of our system expect different premium plans from the system. So we will also need sets which store information about plan money and plan start time, plan end time (in short plan duration). Premium details and the premium of the user will create a relationship named the premium duration of the user.

For the admins we will need a set which stores information about the admin which will have attributes like Admin name, Admin ID etc. Admin will be able to add, remove, edit albums and songs in our system. Admin will have total control over the whole system.

Privacy of the users is the main of our system. We will apply an authentication system in our system. So for that, we will need a set which will store credentials of the users. It will have attributes like User email ID, User name, User password, mobile number, etc. Whenever a user registers it will store his credentials in the system and whenever user logins into the system, it will try to match stored credentials and entered credentials.

Users need to make the transaction to purchase the premium. So we will apply a transaction management system. It will need a set which stores transactions in the

system. We will provide our users various types of transactions because users may want to purchase the premium using a credit card, debit card, UPI etc. It will have attributes like Transaction ID, Transaction money, User ID, Company ID, Transaction type (Only online transactions are applicable) etc. Transaction management system will also be related to the production company because the transaction generally will happen between the user and production company.

We will also need to define some functions and some trigger functions for the system's correct and optimized working. For example when users login we will need to define a function which compares entered credentials with the stored credentials. We can apply a trigger function when a non premium user tries to access the premium songs.

2. Noun (& verb) analysis Tables

1. Noun & analysis

Nouns	**Verbs**
User	
Premium User	
Music	
Music language	Music
Music type	
Premium information	Premium songs
Artist	
Production company	
Admin	
Job	
Transaction	
Album	

Album Rating	
Premium time	
Premium detail	
online	System
large collection	Songs
users	User
maximum users	User
minimum resources	System
server	System
user request	Query
store	Database
user information	User
client application register/signup sign in	Authentication
search	Query, function
user profile	Details
Account type	Premium Accounts
update profile	Query
upload	Add
authentication handling	System
Password	Authentication
money transactions	Payments
update user	Query
update account	Query

premium account	Premium
update song	Query
Music Name	Music
past transaction	Past payments
ID	
bhajan	Music Type
Album name	Album
Artist name	Singer of Music
production company name	Production company
Plan Price	Money
User id	User
User name	User
Email ID	User
Mobile Number	User
edit	query
Count album	Songs in a album
Album id	Artist’s Album
Profit	Company’s profit
Artist ID	artist
Company ID	Album’s Company
Music ID	Music’s Album
query optimization	Optimization
Type	

premium songs	Premium
views	Music
likes	Music
ratings	Music
different premium plans	Plan details
plan money	Money
plan duration	Starting and ending time of a plan
premium duration	Plan duration
Admin id	Admin
edit albums	Query
total control	Admin
privacy	feature
authentication system	System
store credentials	database
user registers	Registration
transaction management system	system
credit card	Transaction Type
debit card	Transaction Type
Transaction ID	Payment details
online transactions	Mode
large collection	collection
appropriate app	application
web / android / desktop	platform

efficient database design	Designing
stream song	streaming
basic functions	functions
workflows	working
client application register/signup	
various filters	
music player	
json	format
postgresql	Tool
ui/ux	User interface
offline availability	
copyright	
omms	System
software	
singer names	
Plan Duration	
different users	
server authentication handling	
store information	
produced	
production companies	
Company name	
main component	

different languages	Languages
trigger function	
Plan Name	
Total song made	Artist
Base salary	Job
Admin Name	Admin
Plan ID	Premium plans
Transaction Mode	Transactions
Sender Type	Transactions
ReceiverType	Transactions

2. Accepted Noun and Verbs list

Candidate entity set	**Candidate attribute set**	**Candidate relationship set**
User	User ID	Authentication
	User Name	
	Password	
	Email ID	
	Mobile Number	
	Account Type	Premium user
	Plan ID	
Artist	Artist ID	Authentication
	Artist Name	

	Password	
	Total song made	
	Company ID	Job
Album	Album ID	Artist's Album
	Album Name	
	Artist ID	
	Album Rating	
	Company ID	
	Profit	Companies Profit
Production Company	Company ID	
	Company Name	
	Base salary	Job
Music	Music ID	
	Music Name	
	Music Language	
	Music Type	
	Artist ID	Singer
	Album ID	Music's Album
	Company ID	
	Premium Information	Premium songs
	Views	
	Likes	
Admin	Admin ID	Authentication
	Admin Name	
	Password	

Premium Details	Plan ID	
	Plan Name	
	Plan Duration	
	Plan Price	
Transaction	Transaction ID	Payment Details
	Transaction Mode	
	Sender ID	
	Receiver ID	
	Sender Type	
	Receiver Type	

3. Rejected noun and verbs list

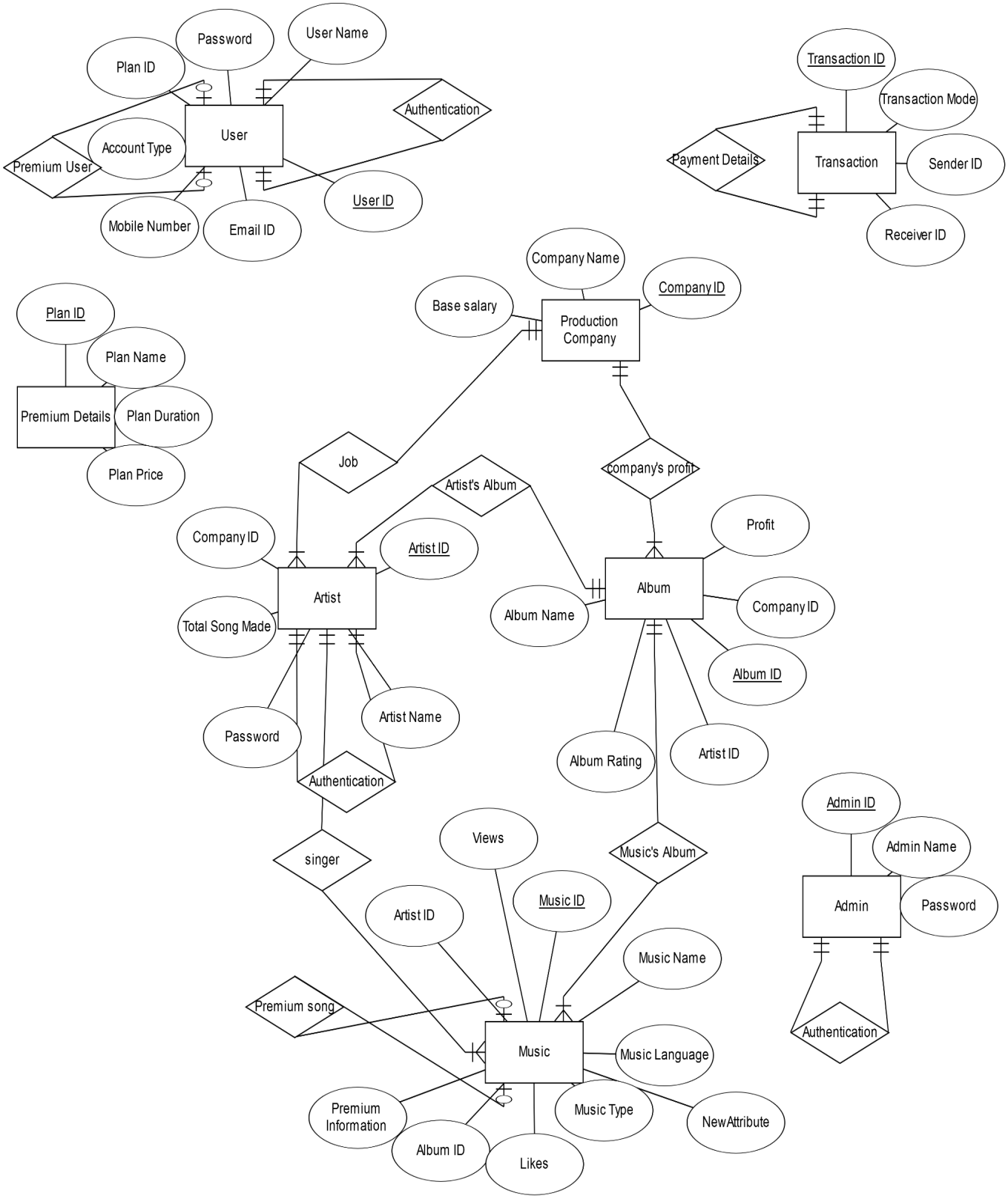
Noun	Reject Reason
trigger function	**Vague**
different languages	**General**
main component	**General**
company name	**Duplicate**
production companies	**Duplicate**
produced	**Vague**
store information	**General**
server authentication handling	**General**
different users	**Vague**
ID	**Attributes**
singer names	**Duplicate**

software	**General**
omms	**General**
copyright	**General**
Type	**Attributes**
offline availability	**Irrelevant**
ui/ux	**Irrelevant**
postgresql	**General**
json	**Irrelevant**
music player	**General**
various filters	**General**
client application register/signup	**General**
workflows	**General**
basic functions	**Vague**
stream song	**Irrelevant**
efficient database design	**General**
web / android / desktop	**General**
appropriate app	**General**
online transactions	**Duplicate**
debit card	**General**
credit card	**General**
transaction management system	**General**
Premium time	**Duplicate**
online	**General**
large collection	**Vague**
users	**Duplicate**

maximum users	**Duplicate**
minimum resources	**General**
server	**General**
user request	**General**
store	**General**
user information	**Duplicate**
client application register/signup sign in	**Duplicate**
search	**General**
user profile	**Duplicate**
update profile	**General**
upload	**General**
authentication handling	**General**
money transactions	**General**
update user	**Duplicate**
update account	**Duplicate**
premium account	**Duplicate**
update song	**General**
past transaction	**Duplicate**
bhajan	**Vague**
production company name	**General**
edit	**General**
Count album	**Duplicate**
query optimization	**General**

premium songs	**Duplicate**
ratings	**Duplicate**
different premium plans	**General**
plan money	**Duplicate**
plan duration	**Duplicate**
premium duration	**Duplicate**
edit albums	**General**
total control	**Irrelevant**
privacy	**Association**
authentication system	**Duplicate**
store credentials	**General**
user registration	**Duplicate**

3. Version 1 of ER diagram



4. Version 2 of ER diagram and final ER diagram