Bilkent University

Department of Computer Engineering

CS 353 - Database Systems

HeyListen: Music Playing Database System

Project Proposal

Group 16

Arkın Yılmaz	21502080	Section 2
Anıl Erken	21501468	Section 2
Berk Mandıracıoğlu	21501741	Section 1
İrem Ural	21502278	Section 2

Course Instructor: Özgür Ulusoy

Project Proposal

Feb 26, 2018

TABLE OF CONTENTS

1. INTRODUCTION

In this report, HeyListen, music playing database system is described by providing some functionalities and by putting some boundaries to the project. Then, as a storage for our application, why and how a database is used in our project is explained. Additionally, functional and nonfunctional requirements and the limitations regarding to the database management system are listed. According to the functionalities of the program, the entity relationship model is given.

2. PROJECT DESCRIPTION

Music playing database system called HeyListen is a web based application, which users can listen and keep track of the songs in more enjoyable and interactive way. This online platform allows users to follow their friends and the singers. Hence there will be followers of the user as well as following ones, which are followed by the user. Users will be informed by the activities of the people they followed. These activities can be as follows:

- Share song with followers
- Like/dislike a song, comment, post
- Follow a playlist, singer or user
- Buy song or album
- Followed by someone
- Join a group
- Make a playlist

These activities will have dates with them and will be shown as friends' activities on the main page. If a followed singer shares a new song, user will see it on their main page as well. They can comment on these posts and like or dislike it. They can share songs. Furthermore, they can create playlists of their own and sharing this playlist is totally up to user. They can keep it private or public. In order to listen the songs or to add them to their playlists they should first buy the songs. Additionally, users can create a group, hence they will become admin of that group. Admin can invite users to the group and the group can create a playlist together. However, only admin can add these songs to the playlist.

From the search engine, users can search the songs by their name or they can find singers from there. Differently from users, the singers in the system will have bios, where their information will be shared. It will include their birthdays, the place of birth, number of sold albums and songs in the platform will be displayed. By using the sold albums and the songs and the number of listeners in a monthly bases, trending songs will be suggested to users. These suggestions may be regional or worldwide. The regional trending songs will be displayed, if the user enters the country, where s/he lives. Additionally, there will be suggestions for each type of music like RnB, pop, rock etc.

2.1. Why a database management system is used for HeyListener?

HeyListener stores huge amount of data about the singers, songs and the users. As it provides several functionalities it requires several relations which should be stored on database systems. For instance, comments are one of the huge and unpredictable data, the texts that are written should be easy to access and store at any time, the database management system will provide these. As vast amounts of information are subject to change and as they are unmanageable in other storage system, we preferred this system. Furthermore, as it is designed to serve for several clients, the user IDs, names and passwords will create an enormous amount of data which cannot be easily managed by file systems. The retrieval and the manipulation of data are much easier in database management systems, and as we would like to create an interactive application, we would like to minimize the response time. Hence, using database management system takes us a step closer to our goal. For searches in the application, this system will ensure faster results than other storage systems. Additionally, as database systems provide several read and write operations at the same time, it will provide availability and scalability for our program. Lastly by creating a good design, an online database system will provide consistent data, the program will be less likely to encounter an error and exceptions in this case.

2.2. How Do We Use Database as a Part of the Project?

Database will enable us to store required information about users, singers, songs, albums, playlists etc. and relationships between them. Firstly, database system will be used for sign-up and login processes to verify the entered credentials. After that, database will provide necessary functionality by presenting the songs, albums and singers to the user. Then, the user will interact with those entities and his/her activities will be stored in the database. Those correspond to the relationships between entities in our E/R Diagram. These relationship tables will help us to provide dynamic data to user, like friend activities, created playlists, song statistics and joined groups. Also there will be a search engine for user to search for singers, songs, albums or other users. This engine will directly use database queries. Our database will change dynamically, as new users join, new songs added, new activities occur -like a follow from one user to another-. At the same time, these changes will cause updates in the menus. For example the follow activity will cause changes in main page of the user. It will now also include recent activities of new followed user. Therefore it is similar to a Model – View – Controller architecture where our database constitutes the Model.

3. REQUIREMENTS

3.1. Functional Requirements

Our Music Playing Database System is made for two types of end users: Music Fans and Music Producers. There are many functionalities that will satisfy all music fans with different interests. Also, the music producers can make use of the data provided by our application. These functional requirements are listed below:

3.1.1. Music Fan

- User should sign-up and login to system with a username and password, for all those functionalities below.
- User should be able to listen all music in the application after buying them.
- User should be able to see all albums and songs of a singer in the database. By navigating in that menu, user can also see all songs in a particular album.
- User should be able to create playlists. In order to add songs to playlist, user should buy the songs or albums.
- User should be able to share songs and playlists. He/she should be able to make comments, like or dislike shared playlists and songs.
- User should be able to follow other users, playlists and singers. In this way, user will see any activities about them in his/her main page. Those activities include new songs from the followed singers, shared playlists and songs from the followed users and update in a followed playlist.
- User should be able to keep her profile or playlist private. In this case, s/he will be able to accept or reject any incoming follow requests.
- User should be able to create/join groups. In a group users can share playlists and songs. In this way user can find friends with similar interests and can form a fan base. A group should have at least one admin. Creator of the group is admin by default. Users that are not part of a group can't view the posts that are shared inside the group. Admin can invite friends to join to the group.
- User should be able to search for singers, users and albums via search engine.
- User should be able to see a short bio of each singer.
- User should be able to see the list of popular songs. This list can be filtered based on location (worldwide/nearby) and genre(all/a specific genre).
- User should be able to see album and song suggestions based on genre.
- User should be able to see statistics about songs, albums, playlists and singers. Those statistics include information about number of listens and number of sells.
- User should be able to buy a song or album for another user.

3.1.2. Music Producer

- User should sign-up and login to system in the same way with a music fan type of user.
- Music producers have the same authorization type with the music fan user type. Therefore they should be able to view the same content. This means they should be able to see statistics about songs, albums and singers. They should also be able to view information about popular songs based on location, date and genre. They can make use of these data in their production process and decisions.
- User should be able to contact with the application maintainers and request the addition of their songs and albums to application.

3.2 Non-Functional Requirements

3.2.1. Reliability

- The system should perform the necessary functions with an acceptable mean time to failure and should detect specified faults and bugs.
- The system should be robust. It should function correctly despite the invalid user inputs.

3.2.2. Performance

- The system should quickly react to user inputs.
- The system should not be down. It should be available and accessible when required for use.
- The system should accomplish all desired tasks by the user within an acceptable time interval.
- The system should be able to handle big amounts of users and data without performance problems.

3.2.3. Usability

- User should easily be able to learn how to use the application with its all functionality.
- User interface elements like colors and fonts should be clean and neat. The menus should be easy to navigate.

2.2.4. Supportability

- Even after the deployment, implementing additional functionality and new entities and relations to system should be easy.
- After the deployment, switching between different technologies should be easy to implement.

3.2.5. Security

- Since the application requires the user to sign-up, it should store the passwords in a safe way.

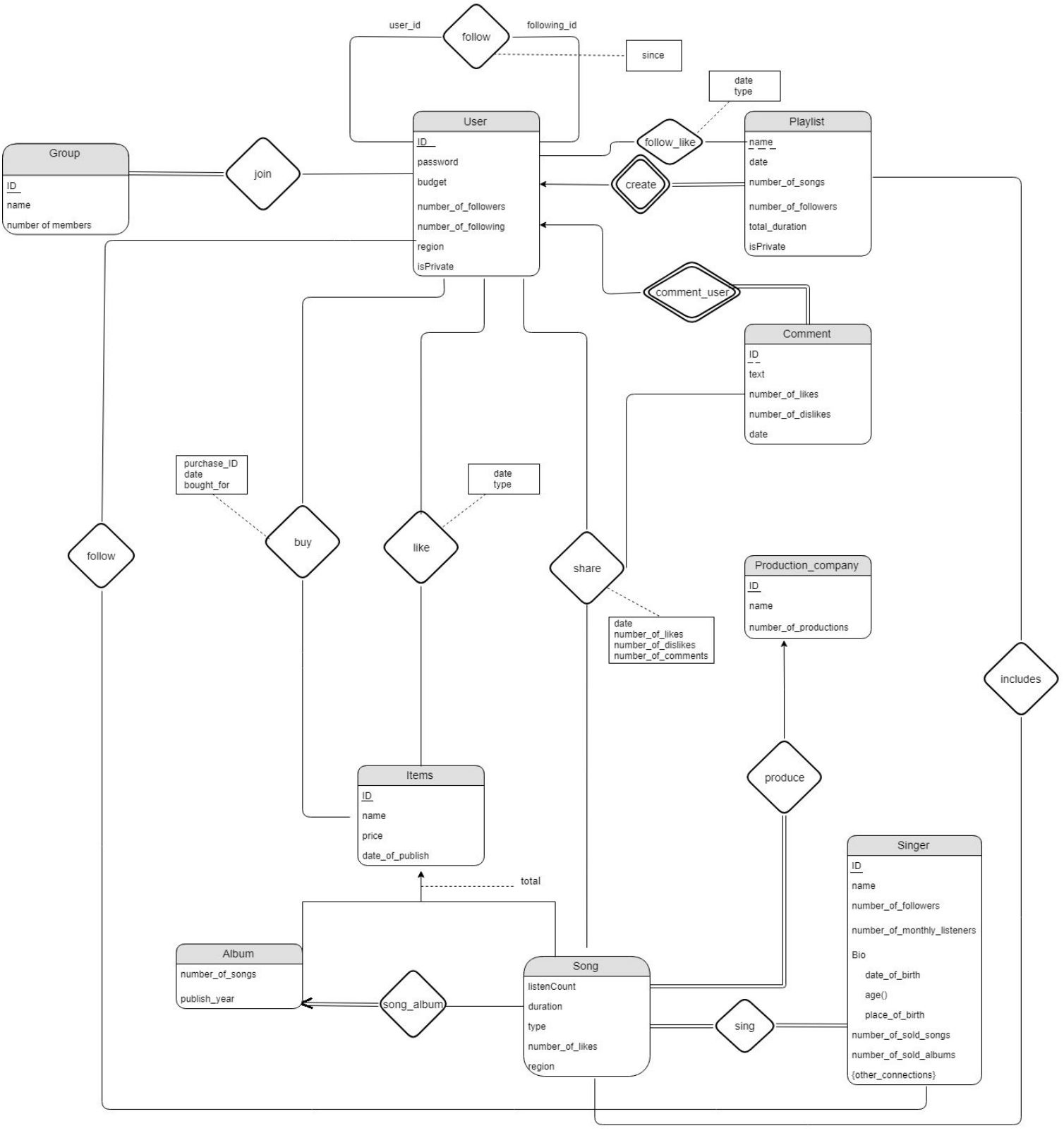
- To guarantee the strength of the passwords, the system should put limitations about the length, usage of special characters etc.

- Since users will be able to buy songs and albums, those processes should be handled in a secure way. Moreover, budgets of the users should be stored safely.

4. LIMITATIONS

- A playlist that is created by a group can only be edited by the group admin.
- Users can not make a playlist out of the songs that they didn't buy.
- Users can't see the activities of users that they don't follow.
- Users can't edit the playlists that were not created by them.
- Only the system admin can add new songs to the system.
- Users can't delete the comments of other people, only the playlist creator can delete them.
- Only the system administrator can delete users from the system.
- A user password must be longer than 6 characters. It must include at least one capital letter and one special character.
- Users that are not part of a group can't view the posts that are shared inside the group.

5. E/R DIAGRAM


follow
user_id
following_id
since
User
ID
password
budget
number_of_followers
number_of_following
region
isPrivate
follow_like
date
type
Playlist
name
date
number_of_songs
number_of_followers
total_duration
isPrivate
create
Group
ID
name
number of members
join
comment_user
Comment
ID
text
number_of_likes
number_of_dislikes
date
purchase_ID
date
bought_for
buy
date
type
like
follow
share
date
number_of_likes
number_of_dislikes
number_of_comments
Production_company
ID
name
number_of_productions
includes
produce
Items
ID
name
price
date_of_publish
total
Singer
ID
name
number_of_followers
number_of_monthly_listeners
Bio
date_of_birth
age()
place_of_birth
number_of_sold_songs
number_of_sold_albums
{other_connections}
Album
number_of_songs
publish_year
song_album
Song
listenCount
duration
type
number_of_likes
region
sing

6. CONCLUSION

HeyListen will be a web application which will be used by many users for music sharing and discovering purposes. It will serve as a music social media / library. Moreover, users will be able to learn the information related to the singer that they are listening. It will be the musical platform that people needed for a long time.

HeyListen will have features that other music platforms mostly lacked. For example, users will be able to comment and make suggestions to playlists and even create social groups. In these groups users will be able to meet with new people that have a similar taste in music.

Detailed information about the project can be viewed in the Project Description where the purpose and the capabilities of the platform are described. Features and functionalities of HeyListen are described in Requirements section that shows what the users should expect from our application. The boundaries that shouldn't be crossed by the users and the system are defined in Limitations section.

7. WEBSITE

https://berkmandiracioglu.github.io/HeyListen/

8. REFERENCES

- Object-Oriented Software Engineering, Using UML, Patterns, and Java, 3rd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2010, ISBN-10: 0136066836.