Project Proposal

Music Playing Database System

(MUZIKA)

Group 12

Alba Mustafaj - 21500009

Argert Boja - 21503525

Rubin Daija - 21500010

Ndricim Rrapi - 21500342

Table of Contents

1. Introduction

Our project consists in developing a database system for a Music App. There are different user activities such as data insertion, organization and retrieval. Therefore, due to the considerably large amount of data which should be stored and organized, using a database, which provides with a simpler and better content management, is the optimal solution. Data can be accessed, altered and updated efficiently according to user preferences. Our music app has two types of users: artists and consumers. These users have different access rights in the app. They can use the app for free, however the consumer has to purchase songs in order to listen to them. In order for a user to sign up, they need to provide the necessary information and then click on the activation link that is sent to them in order to activate and validate their account.

When a consumer buys a song the amount is deposited into the artists account. However, a small percentage is deducted assuming this percentage goes into the company's bank account. Also an artist may not upload songs free of charge. He is also charged a small fee for each new song he publishes into the system assuming this deducted amount again goes into the company's bank account, which is managed by a another money sharing system which we take to be out of the scope of this project. Furthermore, when an artist publishes a song he should put it as part of an album. In case he doesn't do so a random album data is generated. A new album will be automatically created if he uploads a song and does not assign it to any album.

A consumer has several choices: create playlists with songs from several artists, follow artists, connect with other consumers, share songs (and comments) with his/her connections. He/she can buy multiple songs from many artists if the credit left is sufficient. We assume that consumers get free amount of credits when they register which will indicate the payment that they have done through their credit card. Also since this will also be part of an exterior system, consumers' credit will keep incrementing monthly so that they will have the possibility to buy new songs again. However, the sharing process with other consumers only reveals the titles of the songs the consumer is listening to. In order for the other consumers to listen to a song they have to buy it. All the songs purchased by the consumer will be saved in the list of the songs he owns from which he can create playlists and organize them according to his preferences. Furthermore, all the shares of a particular consumer are shown in his wall when his profile is visited. Other consumers can comment on those shared items.

2. Functional Requirements

The database will offer different data manipulation possibilities regarding different user needs, such as adding a song to a library, following a singer, sharing songs and leaving comments. The functions of the database will be as follows:

2.1 Data insertion

- Consumer can create an account
- Artist can create an account
- Consumer can follow an artist
- Consumer can follow another consumer
- Consumer can create a playlist
- Consumer can purchase a song
- Artist can publish a song
- Artist can create an album
- Consumer can leave comments

2.2 Data deletion

- Artist can delete songs
- Artist can delete albums
- Consumers can remove following another consumer
- Consumers can remove songs from playlists
- Consumers can remove playlists
- Artist can delete his account
- Consumer can delete his account
- Consumer can delete comments

2.3 Data modification

- Artists can modify information about their song
- Artist can modify information about their album
- Artist can change general profile information

- Consumer can change account info
- Administrator verifies artist's account

2.4 Data retrieval

- Consumers can listen to songs
- Consumers can look at the description of a song
- Consumers can look at the information about an artist
- Consumers can look at playlists
- Consumers can look at an album's description

2.5 Searching

- Consumer can search other consumers
- Consumer can search songs
- Consumer can search artist
- Consumer can search playlist

3. Non functional requirements

Beside the functional requirements, even the non-functional requirements are a key point in our project. As in every system which operates by using a database, the following non functional requirements are significant assets of our project:

3.1 Security

In order to provide security for our users' data, authentication will be provided through a unique username and password. The system should also not be prone to spam users. This will be provided through an email verification process. In order to secure users data doesn't get deleted accidentally we will make sure to have solid database-injection system, does time permit so

3.2 Performance

In our system, we will try to reduce the query time, response time and processing time as much as possible. Since this project's main aim is that of providing users a pleasant time, spending a lot of time on making computations and queries would not be an advantage. Thus, as developers we will work on having an optimal performance which also helps increasing the users of our music app.

3.3 Integrity

In order to offer services to a myriad variety of users, we will be taking into consideration also how users will access our web-application. In such a case we will make sure we provide compatibility as much as we can so that the users do not experience delays. Furthermore, we will also consider making our web-application mobile-friendly, in order to offer a pleasant experience to users accessing our services from their mobile phones..

3.4 Maintainability

Maintainability is very important for this project since the system might require a lot of maintenance due to the heavy interaction between the users and the system. In order to avoid such situations we will make sure our system is initially efficient and well structured.

3.5 Robustness

In order to create a system which will be easily adapted to future changes, we will also consider robustness as a very important requirement.

Despite the other non-functional requirements we will consider in the future, the above mentioned ones are currently the most important.

4. Limitations

- Users can not register into the system if they do not verify their account through the email verification link.
- A consumer can only view other consumers name, username, publicly shared playlists and date when that user registered. In order to show more information they should follow that user and get followed back as well
- When users choose to delete their account, all of their personal information, songs purchased, playlists created, friends and artists followed will be deleted from the database system. This however doesn't affect the budget of the artist.
- A shared playlist (post) cannot contain more than 10 comments, and each comment cannot have more than 200 characters.
- An artist may upload a single image as their profile picture. This image should not be more than 2MB in size.
- Song names, album names, and playlist names cannot be longer than 40 characters

- Our database system may not contain any Latin script letters (Ç, Ö, Ø, etc…), only the letters of the English alphabet are allowed.
- Usernames should be alphanumeric and cannot contain spaces in between
- A usernames birthdate less than 1900 will be considered invalid input
- Songs uploaded by the artists should not exceed 1MB
- A playlist may not contain more than 150 songs
- Consumers credit and budget may not be less than 0, they cannot make an action if their budget is lower than required.
- User password should be longer than 7 characters and less than 15. The password should contain both numbers and characters
- A consumer cannot access the shared playlists onto another consumers wall unless they follow them back. In an alternative scenario the consumer sharing the playlist can set the playlist share_state to public so that anyone can access them.
- Lyrics for a song cannot be longer than 1000 characters

5. E/R Model

A depiction of our E/R model is provided in Figure1. Each **user** is firstly stored as a **temporary user** until he validates his account. Users are divided into two types through specialization. A user can be a **consumer** or an **artist**. An artist's personal information is recorded together with the number of albums, followers, songs, budget and an image id to maintain their pictures. Artists may create albums which have related information regarding the album as depicted in figure 1. The album contains 1 or more **songs** which have the related information about a particular song.

On the other side the **consumer** entity holds the personal information for that user as shown in figure 1. A consumer may choose among several options. They may purchase songs using their credit. A purchased song means the artist’s budget will be increased accordingly. After the purchase the song is now owned by the consumer and he/she may listen to it. Consumers then can construct playlist with several songs. A consumer may follow other consumers as well. Alternatively, a consumer may choose to share a playlist onto their wall. The shared playlist acts as a post with a title and time when it was shared. Another consumer may view this post and leave a comment regarding the playlist with the prerequisite of following them first. However, in this case the consumer would not be able to listen to playlists their friends shared, if they didn't purchase the songs contained in those playlist. They can view the songs and then decide whether they want to purchase them or not.

Furthermore, consumers may choose to follow artists as well. In this way they will be notified when that particular artist uploaded a new album.

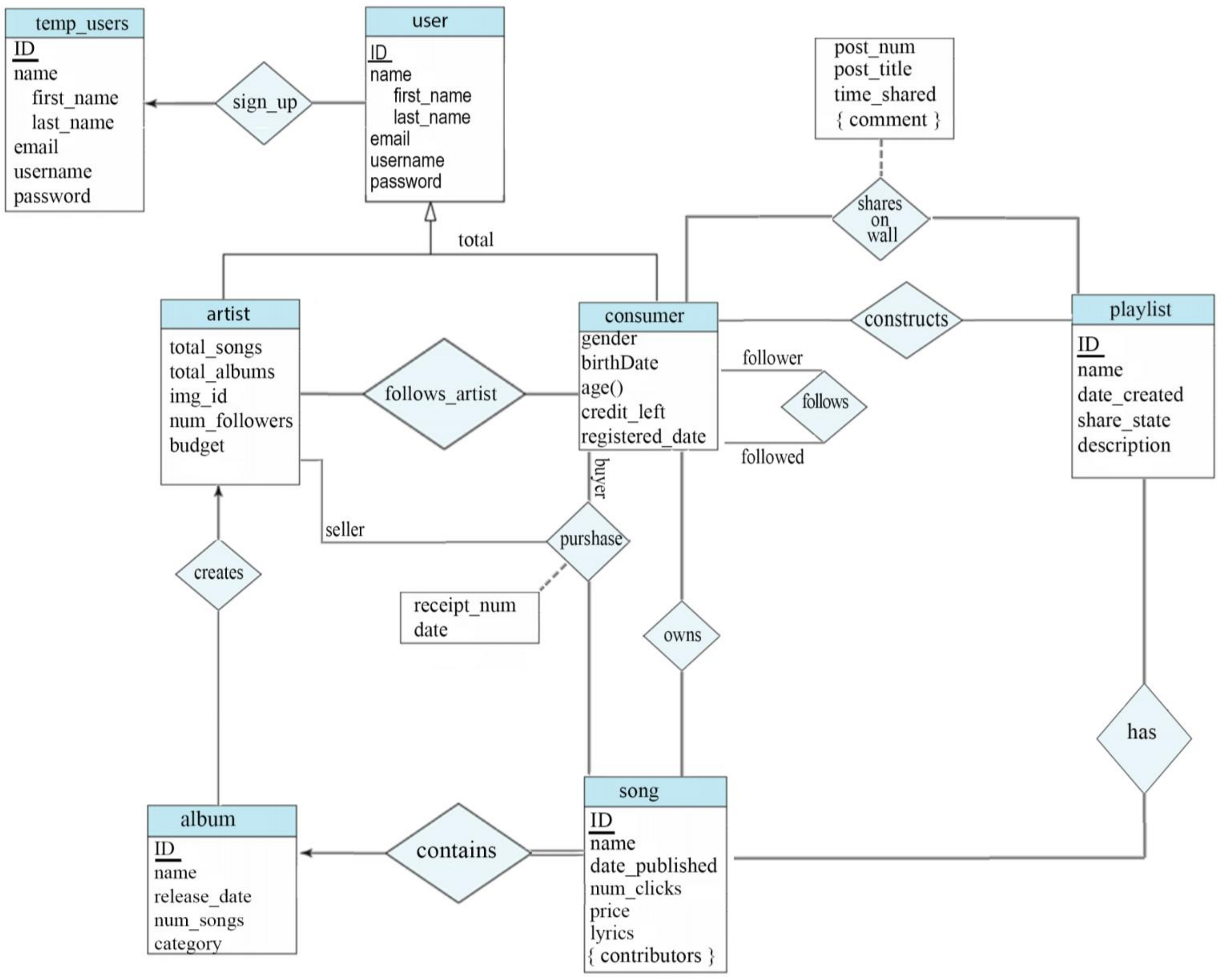

Figure 1-E/R Model

6. Conclusion

All in all, this proposal focuses on clarifying the database system model, functional and non-functional requirements, which are essential for the further steps of development. We aim to build an entertaining app which provides the users with sharing and accessing music from all genres, creating playlists and connecting consumers with each other and artists. The most liked songs will be collected in a playlist based on the number of clicks they have received. It is important that users have an enjoyable and practical music application. In order to help in achieving this goal we have added several additional entities/attributes to our E/R model that will enable users with a variety of options to choose from.

Certainly, a bad database design would directly affect the user's experience. Therefore, our next priority will be optimizing the design for enhancing the performance. Since a part of the users experience is also based on the application's interface, we will work to obtain a nice and enjoyable UI, avoiding any features that may be misleading.

7. Website

The updates, news and reports of this project can be accessed from our webpage:

http://dijkstra.cs.bilkent.edu.tr/~argert.boja/cs353/