Bilkent University

CS353
DATABASE SYSTEMS

PROJECT PROPOSAL REPORT

Ali Bulut - 21402408 - Section 2
Kerem Ayöz - 21501569 - Section 1
Musab Erayman - 21401025 - Section 3
Ömer Faruk Karakaya - 21401431 - Section 3

26.02.2018

Contents

1 Functional Requirements

1.1 Buying a Music

Each user in Nemo will be able to buy music from the market system of Nemo. Any user can purchase any single music as a user can afford to buy it. When the user buys the single song, this song will be added to user's song library and a user can listen to this song anytime.

1.2 Buying an Album

Each user in Nemo will be able to also buy an album from the market system of Nemo. Any user can purchase any album as the user can afford to buy the album. When the user buys the album, songs of this album will be added to user's song library and also the album itself will be added to the user's owned album part. The user will be able to listen to any song of the album any time when s/he buys it.

1.3 Commenting on Playlist

Every user will be able to comment on a playlist and if the playlist is publicly opened everyone, anyone can see the comment. If the playlist is private, the only owner of the playlist can see the comment.

1.4 Creating a Playlist

Every user can create a playlist of his/her owned songs and give it a name. Users cannot create a playlist from another user's song library but do it from only his/her song library. Users need to decide on the creating stage whether the playlist will be public or private. Public playlists will be open everyone to be seen, followed, commented on it, shared and listened if the songs are already is owned by other users. Private playlists will be open to only the owner.

1.5 Rating a Playlist

Users will be able to rate a playlist like a comment. However, the difference is rating a playlist is giving an integer point from 0 to 5. Users will be able to rate playlist which they own if the playlist is private. A public playlist is open to everyone to be rated.

1.6 Sharing a Playlist

Each user will be able to share any public playlist with his/her followers. Private playlists will not be able to be shared.

1.7 Following a Playlist

Users of Nemo will be able to follow public playlist that has been created by other users. When users follow some playlist, that playlist will be shown among users playlist but the user won't be able to change followed but not own playlists.

1.8 Listening Music Previews

In Nemo music system songs and albums has their prices. Users can only listen to 30-second previews of songs that they don't own. Also, users cannot add songs that they did not buy their playlist and again users can just listen to the preview of songs that they don't own even these songs are in the followed playlist.

1.9 Commenting on Song

In Nemo music system users will be able to comment on the songs that they own. They can also change or delete their previous comments. All comments about a song will be able to display by other users. When user make some comment all users that following that user will be notified through their timeline.

1.10 Rating a Song

In Nemo music system users will be able to rate songs that they own from 1 to 5. They can also change their previous rate. Average rating that some song get will be displayed by all users.When a user rates some song all users that following that user will be notified through their timeline.

1.11 Sharing a Song

In Nemo music system users will be able to share songs that they own with their followers. All users who follow that users will be notified through their timeline. While a user sharing some son they will be able to post a comment on it.

1.12 Replying to Comment

In Nemo music system users will be able to reply comments that made to songs and playlists. Users can only reply comments about a song that they own. When a user replies some comment this reply will be displayed with the original comment. When a user replies some comment, all users that following that user will be notified through their timeline. Also, the owner and followers of the owner of the comment which has been replied by other comments will be notified through their timeline their timeline.

1.13Following Users

Nemo allows users to interact with other people very easily. Users could find their friends and followers on the Nemo platform. They could see their friends' activities, popular music, playlists etc. Therefore following other users is a really convenient feature.

1.14 Displaying User/Artist Profiles

One of the best features of Nemo is the dynamic social platform of it. Each user has a unique profile that represents him. Popular songs, playlists, followers of the user could be seen from the profile of the user. Also, users could add their daily favorite songs to their profiles. Coming with these dynamic features and more, in Nemo artists has profiles also. Users could follow artists as like their friends. They are able to follow artists' favorite songs, playlists and more.

1.15 Adding an Album

Artists in Nemo are able to add new albums to the platform. They share it on their profile so that the users could buy their album.

1.16 Adding Money to Wallet

Users could add money to their wallets to buy songs, albums. All they need to do is sending a request to the admin with the card information and done.

1.17 Following a Playlist

Nemo lets a user follow others' playlists, experience the other users' music tastes. After following a playlist, a user could access it from his/her profile. A user is able to see the newly added songs to that playlist and changes made by the creator to that playlist. Also, a user could track the comments made to it.

1.18 Displaying Timeline

In Nemo, every user has a timeline where he/she can view the activities of friends, newly published songs, and albums, shared playlists and comments to them, the rate of their friends to the various playlists and albums and many other features. Thus Nemo is not simply a music platform, it is the new way of interacting with music and people.

1.19 Listening Music (previews included)

Users could listen to the music that they bought from the market. They could add these songs their playlists also. Additionally, they could listen to song previews before buying it, which gives an idea about the music.

2 Non-Functional Requirements

2.1 Operating Constraints

The program will run on the operating system that has a web browser for users. Using on devices that have low storage of RAM could have faced with problems. Also connection quality is important for respond speed.

2.2 Security

Security is a very important issue for applications. In order to protect our data and personal information of users, we will encrypt the password and other related important information.

2.3 Reliability

Since this program is for daily usage for listening music the program has to be reliable. The program should not crash. Users can be able to play what music they want to listen or display any playlist without a problem.

2.4 Performance

In order to respond users' requests quickly, the program has to have high performance to respond and request. Users should not wait long time for their requests. To provide that we will use efficient algorithms in the program.

2.5 Modifiability

In order to compete in the market, the program has to be new features. To provide easily this ability the program could get improvements easily. Modifying a program must not be a big issue.

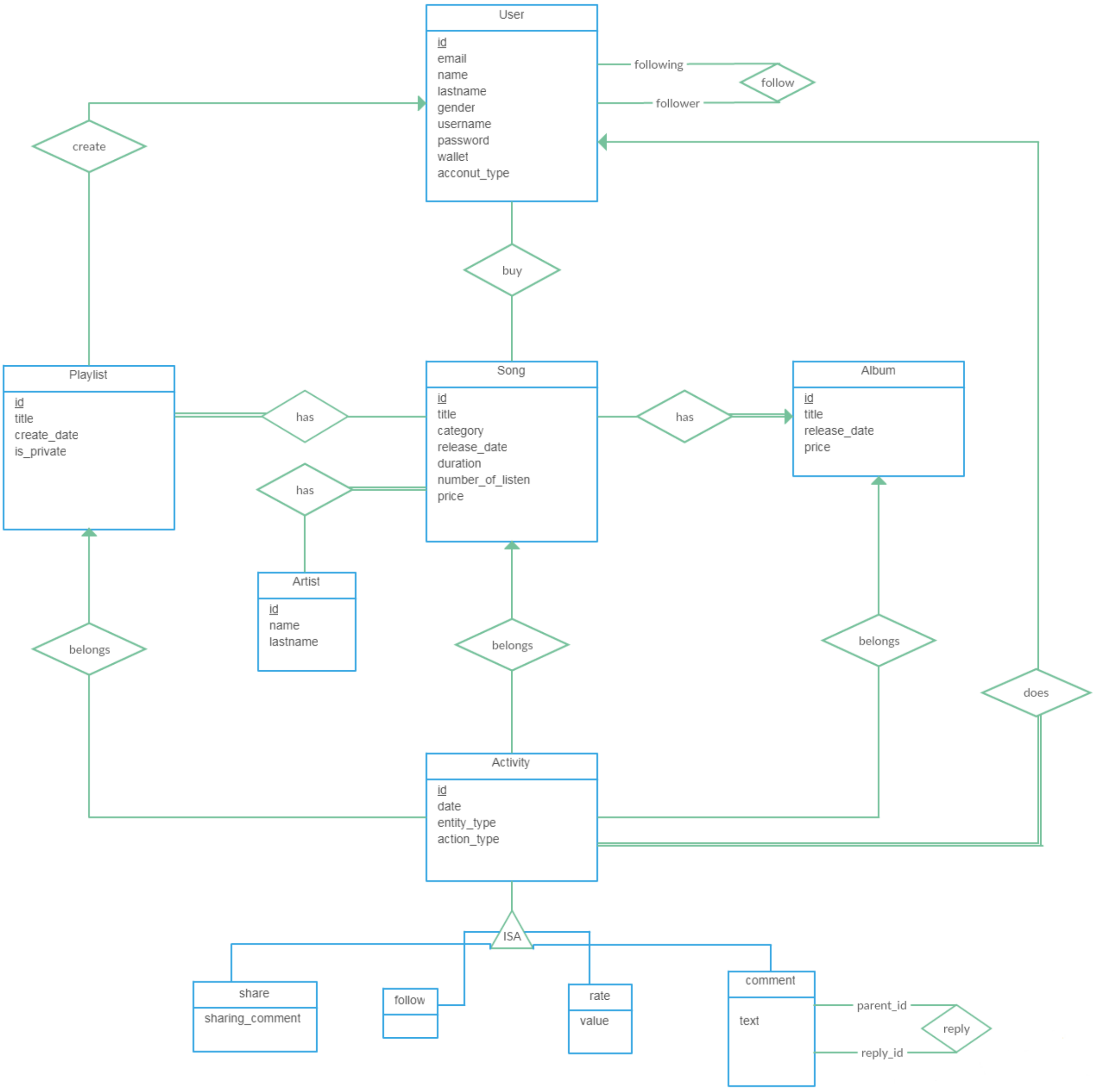
2.6 Usability

The program will have simple, easy to understand user interface. People would not face difficulties while using the program. The program will also be self-explanatory. Understanding what you are doing will be simple and clear.

2.6 Maintainability

The program should be maintainable. A software developer will be able to correct any known defect in a short time. Also, maintenance group must be able to maintain the software. Mean Time to Change (MTTC) for defects should also be as low as possible.

3 ER Design of Database


User
id
email
name
lastname
gender
username
password
wallet
acconut_type
following
follow
follower
create
buy
Playlist
id
title
create_date
is_private
has
Song
id
title
category
release_date
duration
number_of_listen
price
has
Album
id
title
release_date
price
has
Artist
id
name
lastname
belongs
belongs
belongs
does
Activity
id
date
entity_type
action_type
ISA
share
sharing_comment
follow
rate
value
comment
text
parent_id
reply
reply_id