Bilkent University

Department of Computer Engineering

CS 353 - Database Systems

HeyListen: Music Playing Database System

Project Final Report

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 Final Report

April 14, 2018

DESCRIPTION OF APPLICATION **3**

DISTRIBUTION OF WORK **3**

FINAL E/R DIAGRAM **5**

FINAL LIST OF TABLES **6**

4.1 USER 6
4.2 PRODUCTION COMPANY 6
4.3 PLAYLIST 6
4.4 ITEMS 6
4.5 ALBUM 6
4.6 ARTIST 6
4.7 SONG 7
4.8 SONG_ALBUM 7
4.9 HAS_SONG 7
4.10 POST 7
4.11 MUSICIAN_CONNECTIONS 7
4.12 FOLLOW_ARTIST 7
4.13 PLAYLIST_INCLUDES 8
4.14 FOLLOW_LIKE_PLAYLIST 8
4.15 FOLLOW_USER 8
4.16 GROUPS 8
4.17 INVITE_USER 8
4.18 LIKE_COMMENT 9
4.19 SHARE_IN_GROUP 9
4.20 POST_SONG 9
4.21 POST_PLAYLIST 9
4.22 LIKE_POST 9
4.23 LIKE_ITEM 10
4.24 BUY 10
4.25 COMMENT_ON_ITEM 10
4.26 COMMENT_ON_POST 10
4.27 COMMENT_ON_PLAYLIST 10
4.28 COMMENT 11
4.29 HAS_GROUP 11

5.0 IMPLEMENTATION DETAILS **12**

6.0 ADVANCED DATABASE FEATURES **13**

6.1 VIEWS 13
6.2 STORED PROCEDURES/ REPORTS 13
6.2.1 View most sold song type with how many times does it sold 13
6.2.1 View most sold song number with it’s region 14

6.3 TRIGGERS 15
6.4 CONSTRAINTS 16

7.0 USER'S MANUAL 17
7.1 Login 17
7.2 Register 17
7.3 Main Page 18
7.4 Overview 19
7.4.1 Overview (Change Settings) 20
7.4.2 Overview (Add budget) 21
7.5. Playlists 22
7.6 Playlist Page of User 24
7.7. Groups 26
7.8. Group Page 27
6.9 Followings 28
7.10. Followers 29
7.11. Discover 30
7.12. Songs 32
7.13. Albums 33
7.14. Search 33
7.15. Song Page 36
7.16. Artist Page 38
7.17.Production Company Page as User 38
7.18. Viewing Other Users' Profiles 39
7.19. Album Page 40
7.20. Artist User Type 41

1. DESCRIPTION OF APPLICATION

HeyListen is a music playing database system where there are several user types such as guest, musician(artist), and production company.

Guest represents regular users who can create playlists and follow other guests, playlists, artists. Guests can view the activities of the people they are following on their main page. These activities include: following a playlist, friend, and liking, disliking, sharing playlists and songs. Guests should buy songs or albums in order to listen them. User can add budget by their credit card information. Guests can discover new songs in discover page where they can view trending songs and buy them. They can also search other users, songs, albums, singers and production companies. They can create playlists out of the songs they have bought. Users can comment on playlists and they can like/dislike these playlists. Guests can share playlists, songs. Moreover, guests can create new groups to socialize with people that have similar music tastes. In these groups, people can share songs and playlists. Group members can collaboratively create new playlists which demonstrates combined music taste of group members. Musician(artist) can do all the things that a guest can do such as creating a group, sharing playlist and etc. In addition, musicians can add new songs and albums to the database.Production company can view statistical data about the songs they have produced. These statistical data include: number of sells, number of listens, number of follows an etc.

2. DISTRIBUTION OF WORK

Every group member worked on every page of the HeyListen both in terms of SQL queries and PHP code to connect the application logic with the database. EN SON YAZARIZ SPESİFİK OLARAK HERKESİN NE YAPTIĞINI.

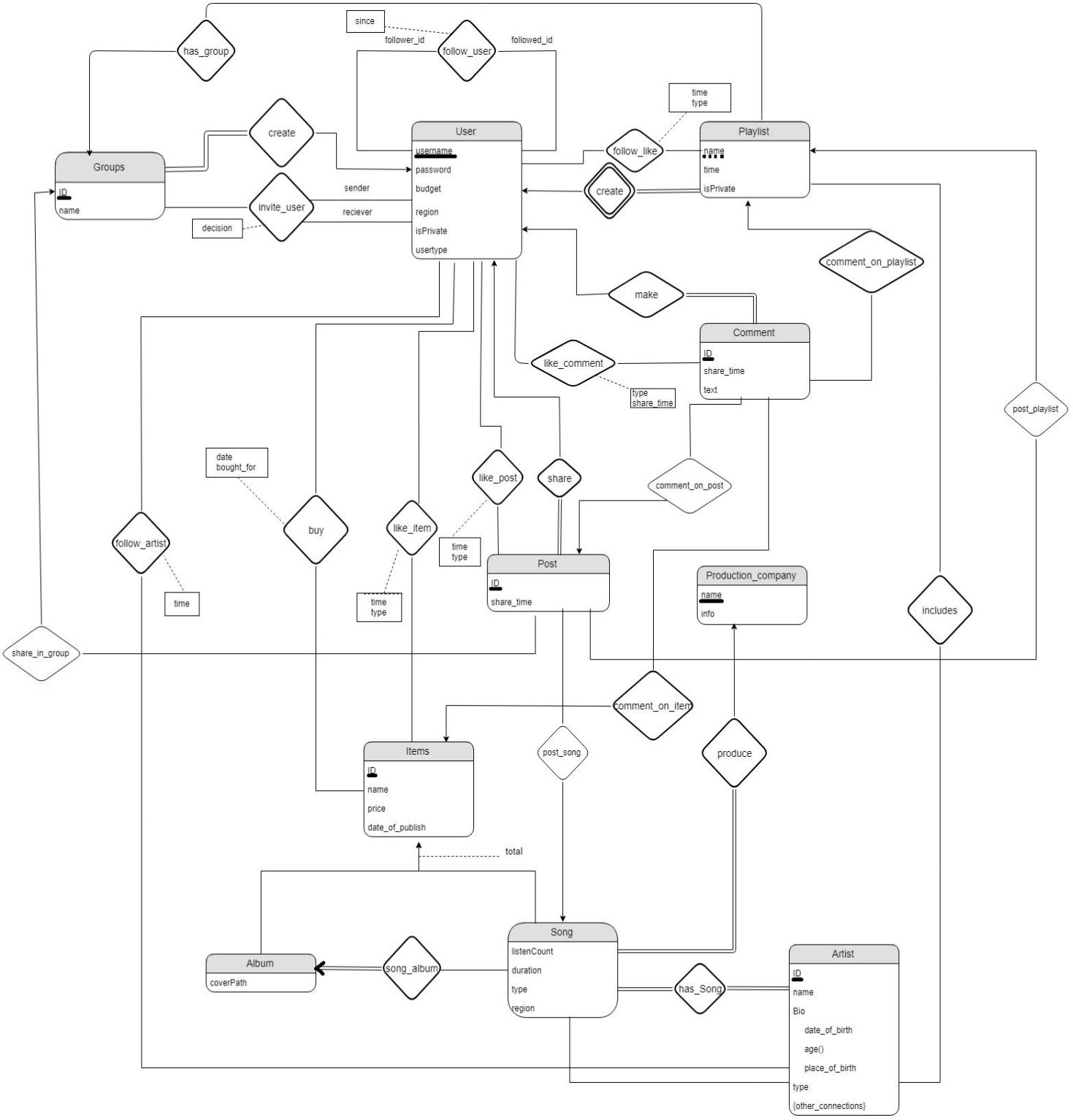
Anıl- worked on login, register, discover and group pages. Also worked on adding a song to a playlist, adding budget with card information, creating/deleting a playlist, creating/removing groups and following/unfollowing other users.

Berk- worked on overview and main page to show friends' and your activity; on comment, playlist, playlists, artist pages; on designing base UIs for all pages and on advanced db components.

Arkın- worked on playlist, playlists, artist, production company, group and album pages and made some improvements on UI.

Irem- worked on search, playlists, playlist, songs, song, follower, following pages, made some improvements on UI, worked on advanced db components

3. FINAL E/R DIAGRAM



4. FINAL LIST OF TABLES

4.1 USER

user (username, password, budget, region, isPrivate,usertype)

4.2 PRODUCTION COMPANY

production_company(name, info)

4.3 PLAYLIST

playlist (username, name, time, isPrivate)
FK: username references user(username)

4.4 ITEMS

items(ID, name, price, date_of_publish)

4.5 ALBUM

album(ID, coverpath)
FK: ID references items(ID)

4.6 ARTIST

artist(ID, name, date_of_birth, place_of_birth, artisttype)

4.7 SONG

song(ID, duration, listenCount, type, region, companyName)
FK: ID references items(ID)
FK: companyName references production_company (name)

4.8 SONG_ALBUM

song_album (albumID , songID)
FK: albumID references items(ID)
FK: songID references items(ID)

4.9 HAS_SONG

has_Song(artistID, songID)
FK: songID references song(ID)
FK: artistID references artist(ID)

4.10 POST

post (ID, share_time, username)
FK: username references user(username)

4.11 MUSICIAN_CONNECTIONS

musician_connections(ID, other_connections)
FK: ID references artist(ID)

4.12 FOLLOW_ARTIST

follow_artist (username, ID, time)
FK: username references user(username)
FK: ID references artist(ID)

4.13 PLAYLIST_INCLUDES

playlist_includes(username,name, songID)
FK: (username,name) references playlist(username,name)
FK: songID references song(ID)

4.14 FOLLOW_LIKE_PLAYLIST

follow_like_playlist (userUsername, creatorUsername, playlistName, type, time)
FK: userUsername references user(username)
FK: (creatorUsername, playlistName) references playlist(username,name)

4.15 FOLLOW_USER

follow_user (follower_id,followed _id, since)
FK: follower_id references user(username)
FK: follow_id references user(username)

4.16 GROUPS

groups(ID, admin, name)
FK: admin references user(username)

4.17 INVITE_USER

invite_user (groupID, reciever, decision)
FK: groupID references groups(ID)

FK: receiver references user(username)

4.18 LIKE_COMMENT

like_comment(liker , ID, share_time, type)
FK: liker references user(username)
FK: ID references comment(ID)

4.19 SHARE_IN_GROUP

share_in_group(post_ID, g_ID)
FK: g_ID references groups(ID)
FK: post_ID references post(ID)

4.20 POST_SONG

post_song(p_ID, s_ID)
FK: s_ID references song(ID)
FK: p_ID references post(ID)

4.21 POST_PLAYLIST

post_playlist(post_ID, p_name, p_username)
FK: (p_name, p_username) references playlist (name, username)
FK: post_ID references post(ID)

4.22 LIKE_POST

like_post(u_username, p_ID, time, type)
FK: u_username references user(username)
FK: p_ID references post(ID)

4.23 LIKE_ITEM

like_item(username, itemID, time, type)
FK: username references user(username)
FK: itemID references items(ID)

4.24 BUY

buy(buyer, itemID, bought_for, date)
FK: buyer references user(username)
FK: bought_for references user(username)
FK: itemID references items(ID)

4.25 COMMENT_ON_ITEM

comment_on_item(commentID, itemID)
FK: commentID references comment(ID)
FK: itemID references items(ID)

4.26 COMMENT_ON_POST

comment_on_post(commentID, postID)
FK: commentID references comment(ID)
FK: postID references post(ID)

4.27 COMMENT_ON_PLAYLIST

comment_on_playlist(commentID, owner, playlistName)
FK: commentID references comment(ID)
FK: (owner, playlistName) references playlist(username, name)

4.28 COMMENT

comment (ID, username, share_time, text)
FK: username references user(username)

4.29 HAS_GROUP

hasGroup (groupID, playlistOwner, playlistName)
FK: groupID references groups(ID)
FK: (playlistOwner,playlistName) references playlist(username,name)

5.0 IMPLEMENTATION DETAILS

In this project, mainly PHP is used as backend, for some pages we need to use javaScript as well. For frontend HTML is used and to build a better UI Bootstrap is used as a framework. MariaDB database in djkstra is used to store data. MYSQL is used to build database and for data retrieval and modification.
Webpage can be found in the following link:

Sometimes it is hard to connect server and client, when a button is clicked or a link is clicked, certain actions should be taken and it can be hard to achieve it. When there is an error in php file, we have hard times to detect them, therefore implementing a page takes a lot of time. Secondly, in order to display user activities on the main page, several results are taken from different queries like follow_user, follow_playlist and like_playlist etc. As there are several results, for each result a php object is created and they are collected in an array and sorted according to their time. As each result is limited by 10, it does not require a huge array.

6.0 ADVANCED DATABASE FEATURES

6.1 VIEWS

The shared songs by the user can be found with the following view, it is used for overview page of the user.

- **Display songs that user shared in her profile**

```
CREATE VIEW songPostOnUserPage( name, duration, type, share_time)  AS
        SELECT I.name, S.duration, S.type , P2.share_time
        FROM post_song P1, items I, song S, post P2
        WHERE I.ID = P1.s_ID  AND  I.ID = s.ID  AND P2.ID =  P1.p_ID
        ORDER BY P2.share_time DESC
        LIMIT 10 ;
```

- **View is used to find which songs are bought by user:**

```
CREATE VIEW A AS select I.ID, I.name, S.type,S.region,I.price from song S, items I
, buy B where S.ID = I.ID and B.itemID = I.ID;
```

This view is used frequently, an example is given in 5.2, it used to check whether a song is bought or not, in pages where is buy option. This view is used in a query.

6.2 STORED PROCEDURES/ REPORTS

In our database, reports are stored as a procedure to access them easier as they are complex queries. These two reports are used in order to view most sold song type and in which region the song is sold.

6.2.1 View most sold song type with how many times does it sold

```
delimiter //

create procedure showMostSoldType3( out count INT ,out typess VARCHAR(45))
begin
select
  max(B.c) into count
from
  (
```

```
   select count(*) as c,
     type as types
   from A
   group by type
  ) as B ;

select B.types into typess
from
  (
   select count(*) as c,  type as types
   from A
   group by type
  ) as B ;

  end;
  //
```

Output:

```
MariaDB [iremural]> call showMostSoldType3(@max,@type);
Query OK, 1 row affected (0.01 sec)

MariaDB [iremural]> select @max,@type;
+------+-------+
| @max | @type |
+------+-------+
|    4 | pop   |
+------+-------+
1 row in set (0.00 sec)
```

6.2.1 View most sold song number with it's region

```
delimiter //

create procedure showMostSoldRegion1( out count INT ,out regionss
VARCHAR(45))
begin
select
  max(B.c) as count
from
  ( select count(*) as c,  region as regions  from  A  group by region
  ) as B ;

select
```

```
  B.regions into regionss
from
  (   select   region as regions  from  A  group by   region ) as B ;

  end;
  //
```

Output:

```
MariaDB [iremural]> select @count2,@region2;
+---------+----------+
| @count2 | @region2 |
+---------+----------+
|       4 | UK       |
+---------+----------+
1 row in set (0.00 sec)
```

6.3 TRIGGERS

1- If the user is registering as an artist then insert that user into artist table with the following trigger:

```
DELIMITER //
CREATE TRIGGER trigArtist
AFTER INSERT ON user
FOR EACH ROW
BEGIN
 IF NEW.usertype = 2 THEN
  INSERT into artist(name,artistType) values (NEW.username,NEW.usertype);
 END IF;
END; //
```

2- When user buys an item and the buy entity is updated, the budget of the user is decremented in the same amount.

```
DELIMITER //
CREATE TRIGGER decreaseBudget
AFTER INSERT ON buy
FOR EACH ROW
BEGIN
```

UPDATE user SET budget = budget - (select price from items where ID = NEW.itemID) where username = NEW.buyer;
END; //

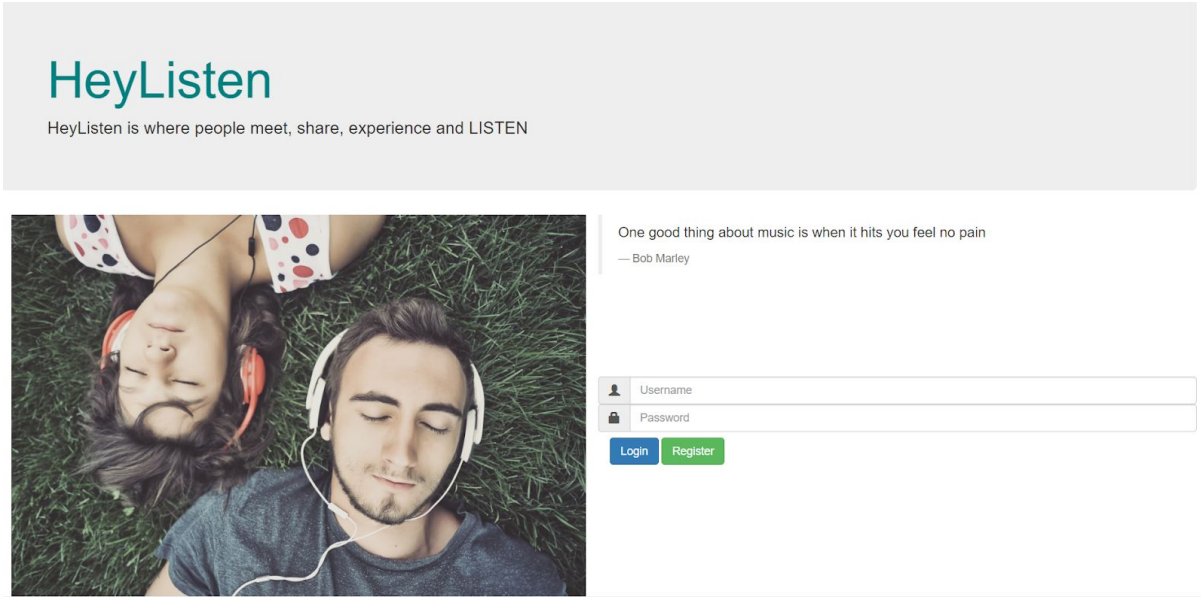
6.4 CONSTRAINTS

1. Users must register to the system to use this web based application. If they already have an account they should first login to the system.
2. User must buy the song in order to listen it.
3. There can be at most 10 posts in the overview pages.
4. Posts can have either song or playlist.
5. One post can be shared in group or in the main page.
6. User cannot post empty posts or comments.
7. For playlist follow_like type attribute can be 1, up to 5.

- 1 for follow
- 2 to unfollow a playlist
- 3 for like
- 4 for dislike

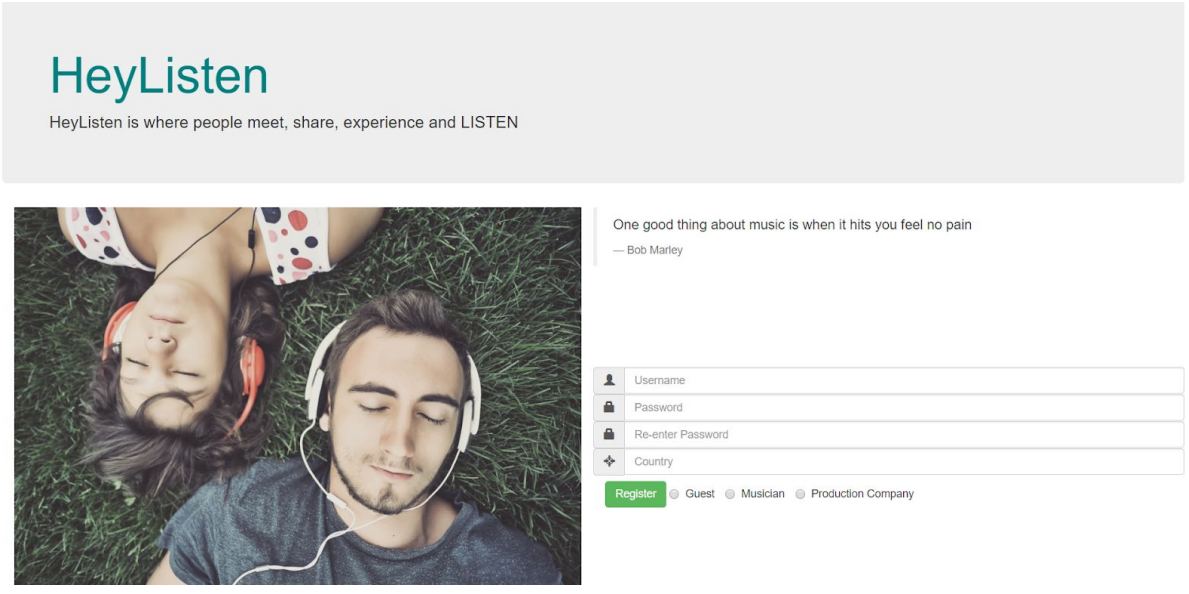
7.0 USER'S MANUAL

7.1 Login



Login page is the first page in the website. User can login by providing correct username and password taken before. Then, he/she will be directed to main page. If user does not have an account, she can click to register button to go to register page.

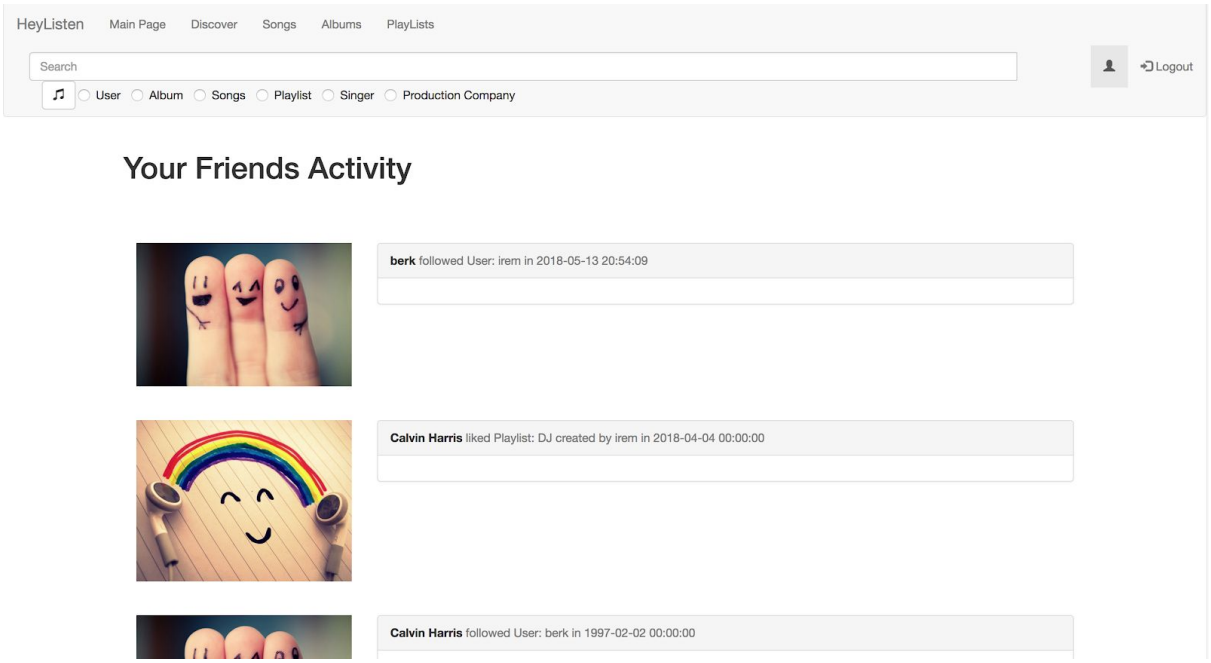
7.2 Register



User can create a new account in this page. User name, password, country and user type must be provided. All fields must be filled, two entered passwords must be equal, password must be at least 6 characters and user name must not be taken before. User must choose

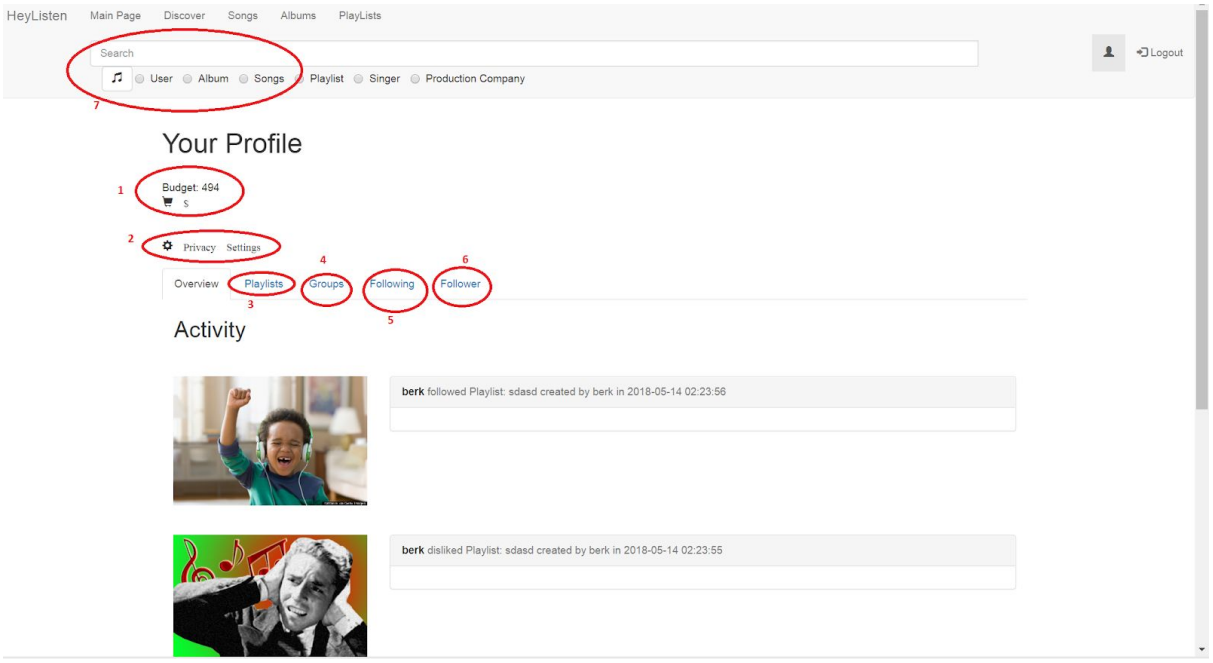
the user type. Guest is for music fans; musician and production company are for music makers. After a successful registration, user is directed to the main page.

7.3 Main Page



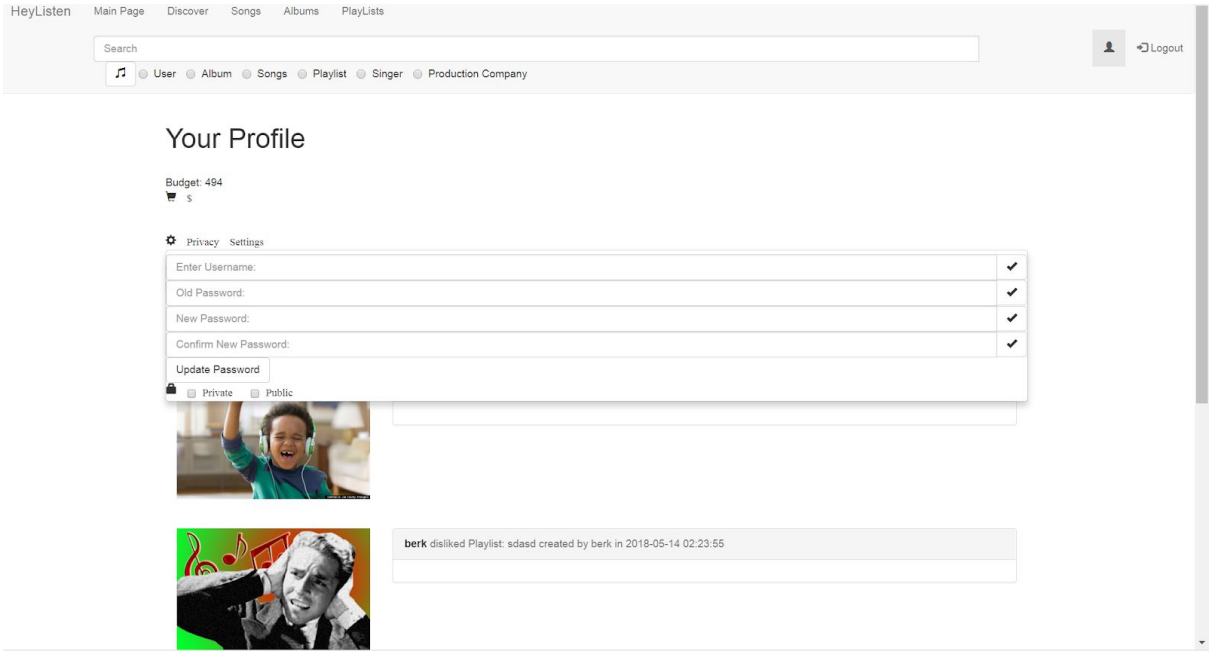
User can see followed users activities on the main page.

7.4 Overview



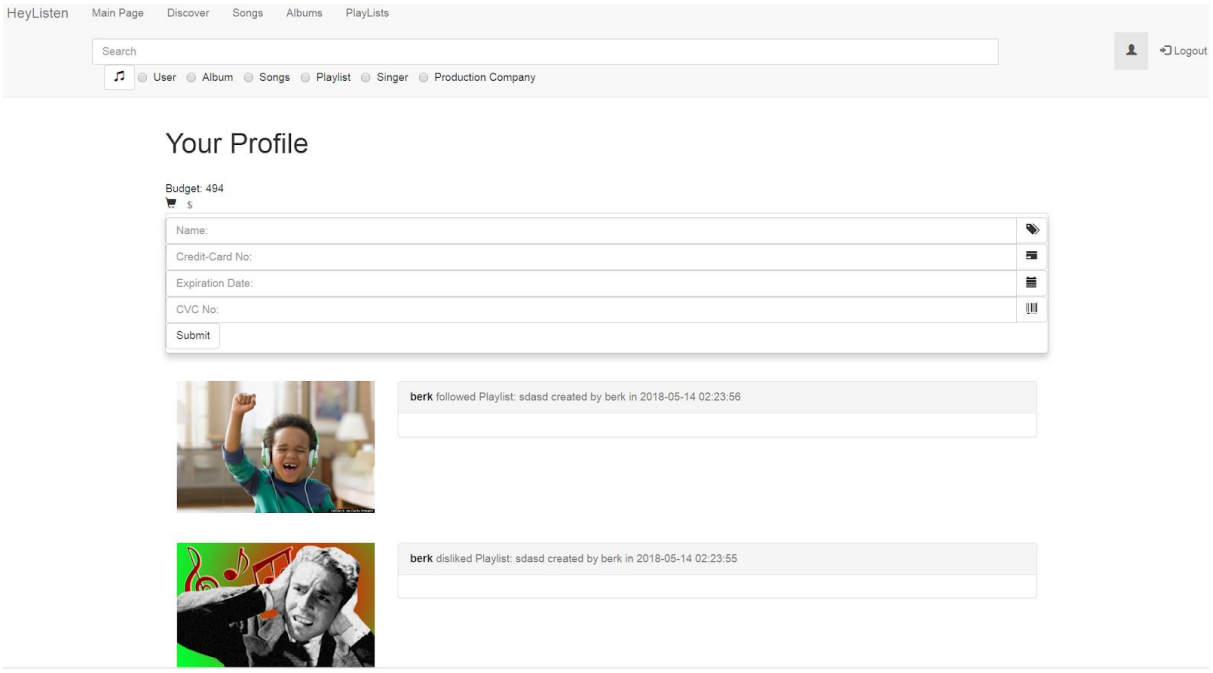
1-User can see his/her budget and by clicking the shopping basket user can add money to his/her account.
2- Privacy settings of the user can be changed, there are two buttons, user should select one and submit it with a submit button next to it.
3- User can see his/her playlists that he/she created, and the lists followed by user.
4- User's groups can be seen from this tab.
5- The followed users/artists can be seen from following.
6- Followers of user can be seen from this tab.
7- User can search users, albums,songs, playlists, singer and production company.
And through the most upper tabs, user can go to Main page, discover etc. any time.

7.4.1 Overview (Change Settings)



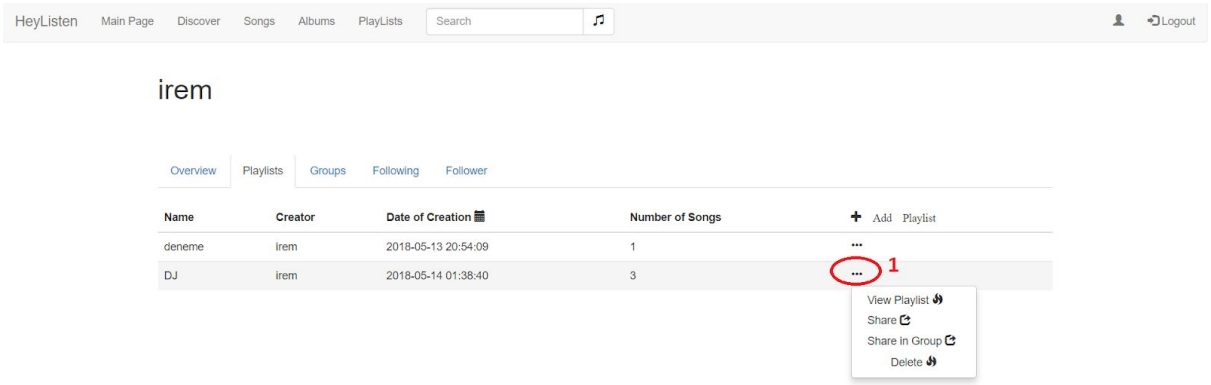
User can change user settings from the above pop-up. Old password should be equal to the new password. And new password should be entered twice. Update password button should be clicked. After selecting private or public user should submit it by clicking to the lock.

7.4.2 Overview (Add budget)

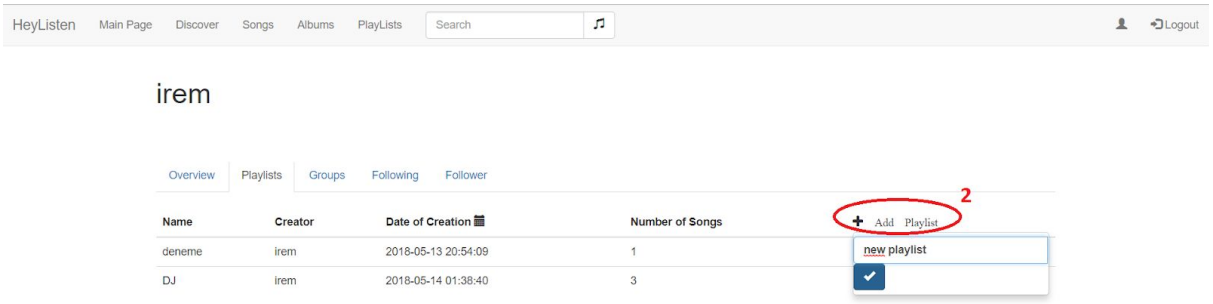


User can add budget by clicking to the shopping basket. User should enter card name, 16 digit credit card number, 4 digit expiration date and 3 digit CVC number then he/she should click submit button.

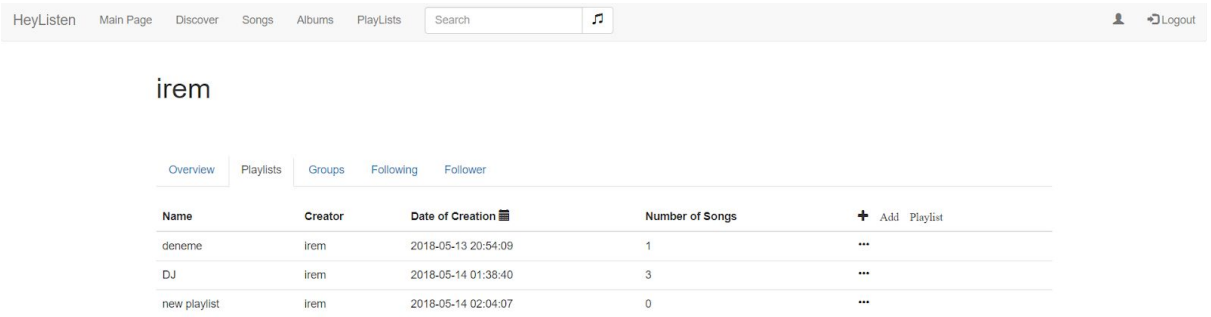
7.5. Playlists



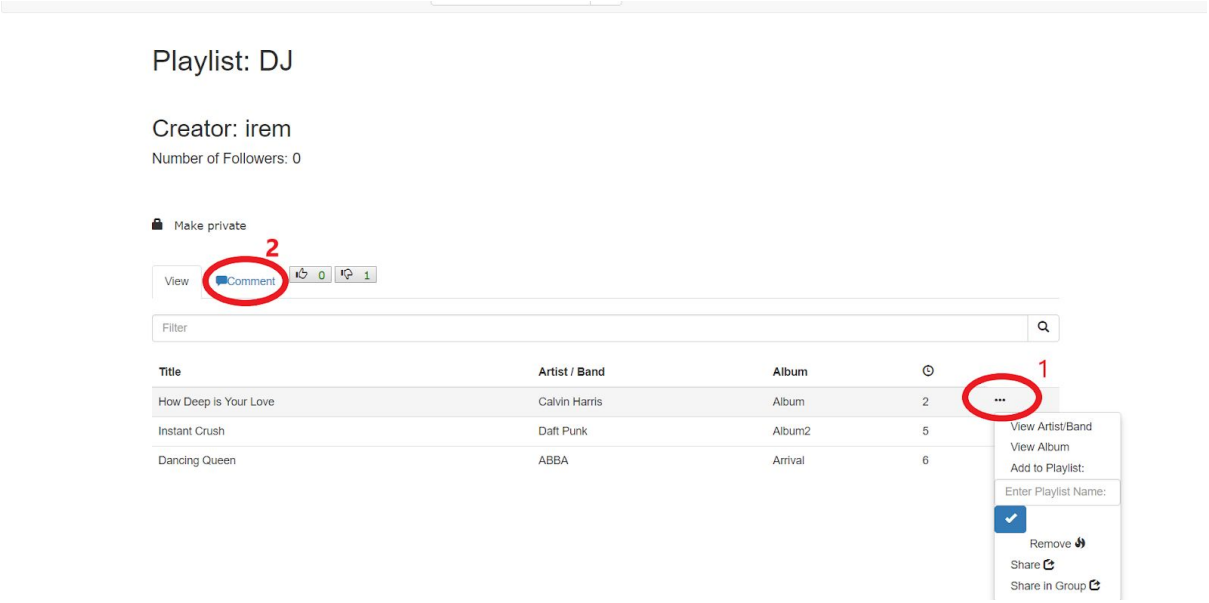
In playlists page, user sees the playlists she created or she is following. Playlist's name, creator, date and number of songs in it are displayed. By pressing the button labeled by "1", a dropdown menu appears. It provides buttons to view the playlist in detail (directs to playlist's page, described at section 6) and to delete the playlist.



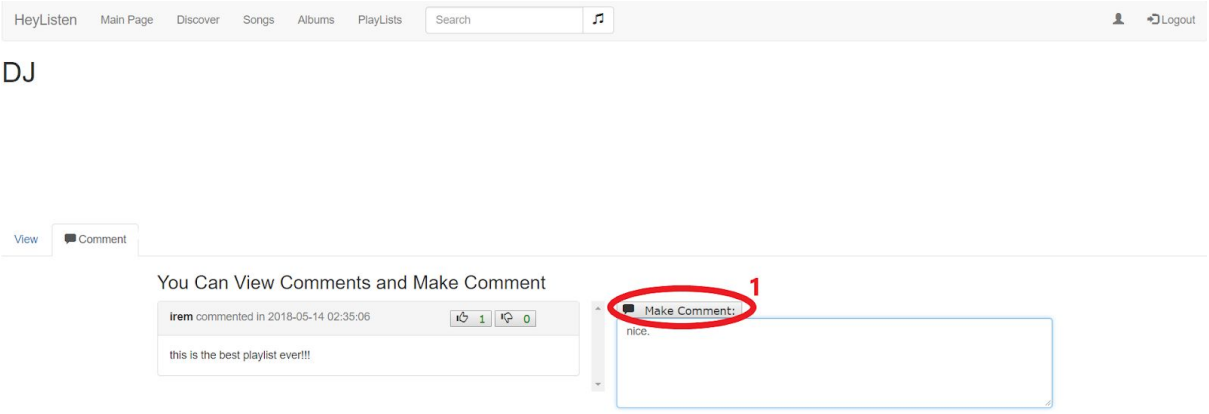
When add playlist button (labeled by "2") is clicked, a text box and a button appears. With this, user can create a new, empty playlist. Here is the new screen after the creation of new playlist:


HeyListen
Main Page
Discover
Songs
Albums
PlayLists
Search
Logout
irem
Overview
Playlists
Groups
Following
Follower
Name
Creator
Date of Creation
Number of Songs
Add Playlist
deneme
irem
2018-05-13 20:54:09
1
DJ
irem
2018-05-14 01:38:40
3
new playlist
irem
2018-05-14 02:04:07
0

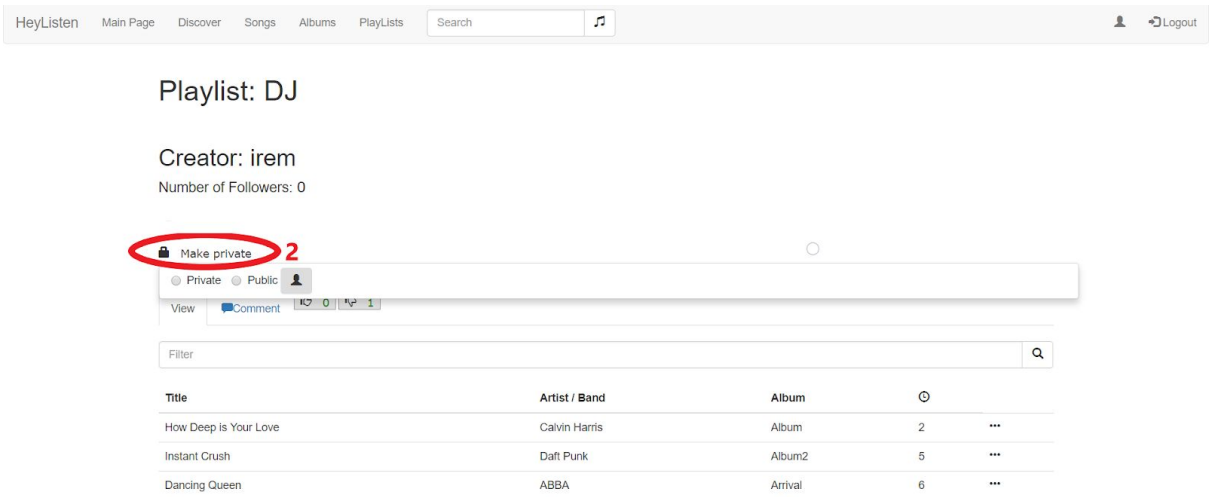
7.6 Playlist Page of User



In this page, user can see detailed information about a playlist. User can see each song's name, artist, album and duration. By clicking the button in "1", user will have options to view artist's page, album's page, add the song to another playlist she created, and remove the song from the playlist. Number of followers of playlist and like/dislike counts are also displayed. By clicking button "2", user can go to comments tab of the playlist.

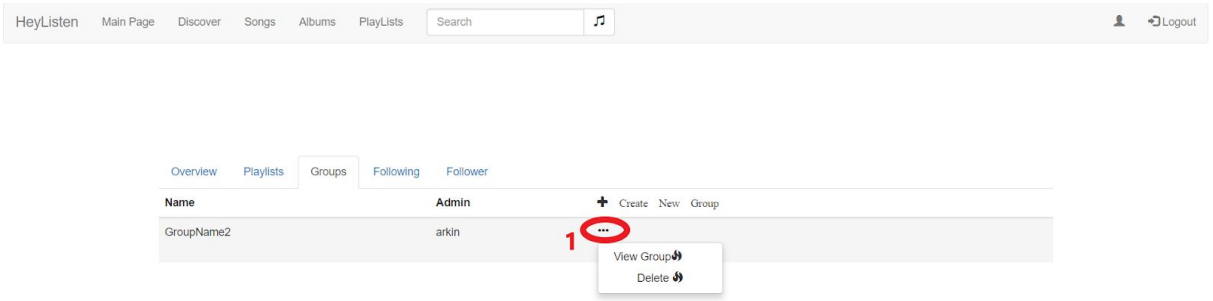


In comments tab, user can make a comment (labeled with "1"), and view previous comments. Each comment has like and dislike options(as seen in left).

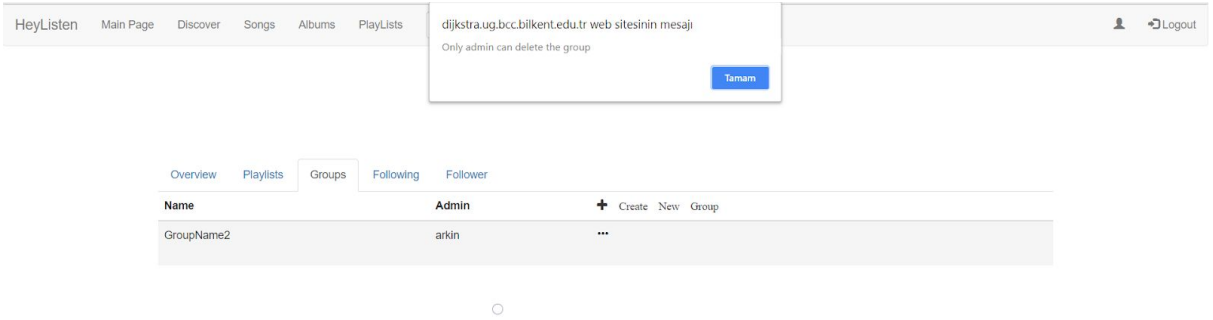


User can change privacy of the playlist by the button labeled with “2”. When it is private, only people who follows the user can view the playlist.

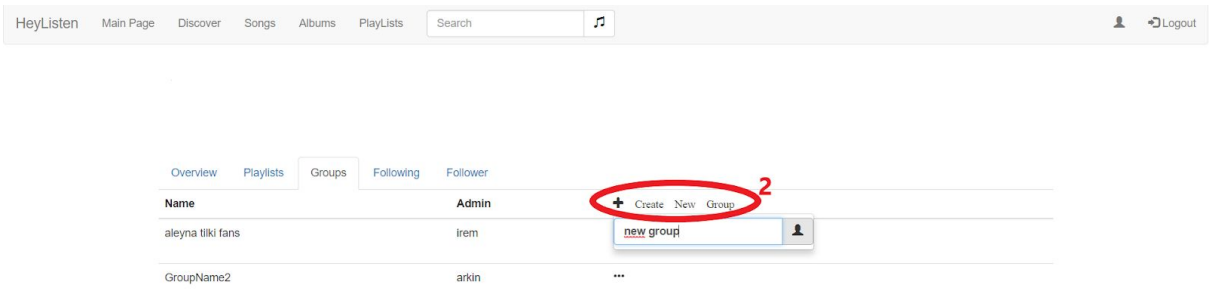
7.7. Groups



In the groups page, user can see the groups she created or joined. Each group's name and admin is displayed. User can view the group (described in section 8) and delete group by button "1". Only admin can delete a group.

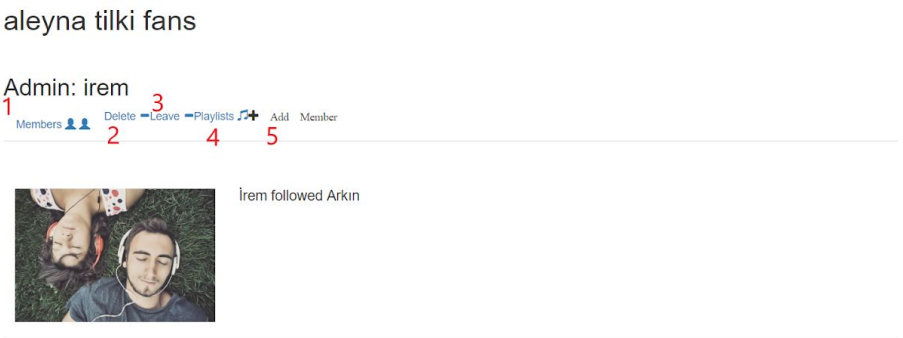


If a user who is not admin tries to delete a group, a warning message is given saying "Only admin can delete the group."

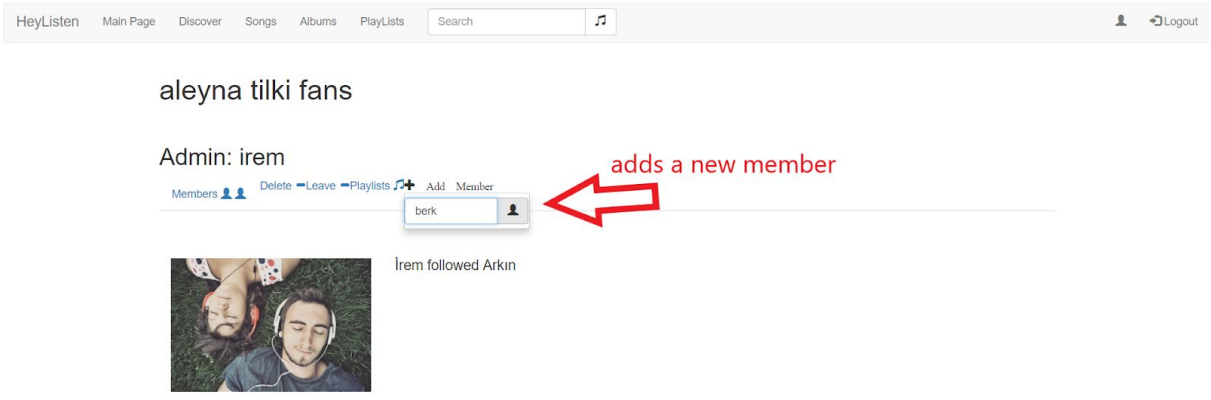


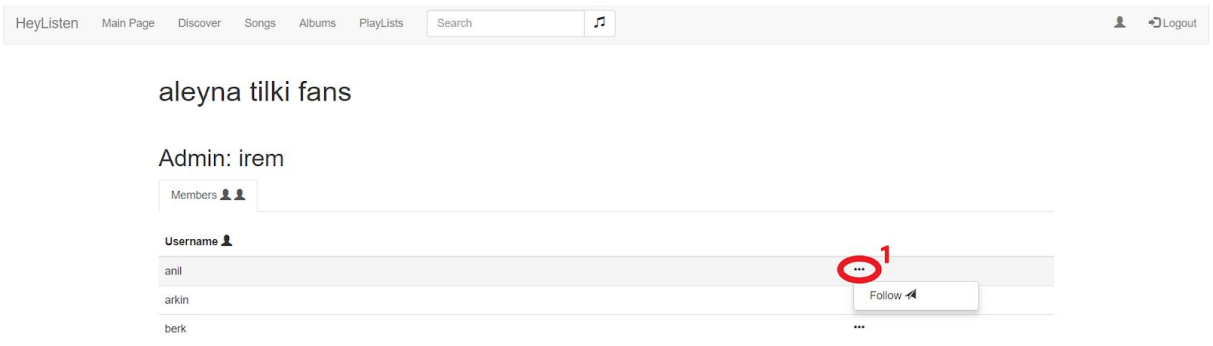
User can create a new group by using the button labeled with “2”.

7.8. Group Page



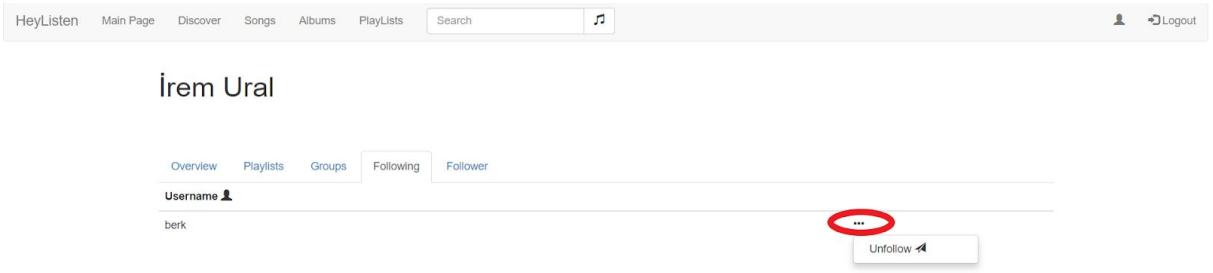
In the group page, activities of members are displayed. Also, a member have options to see members of the group (1), leave the group (3) and see the playlists of the group (4). Admin can also delete the group (2) and add a member to it by entering the username to add (5).





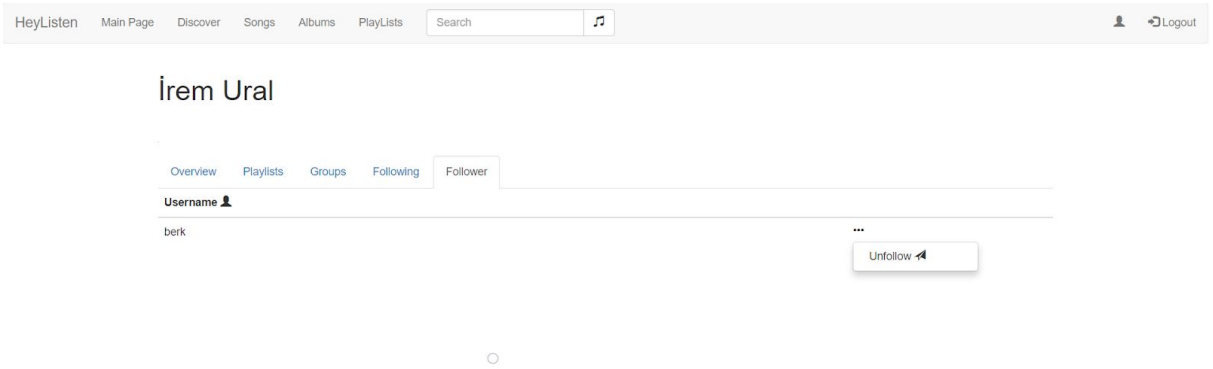
In members tab, group members are displayed. Each member can be followed/unfollowed by using button “1”.??

6.9 Followings



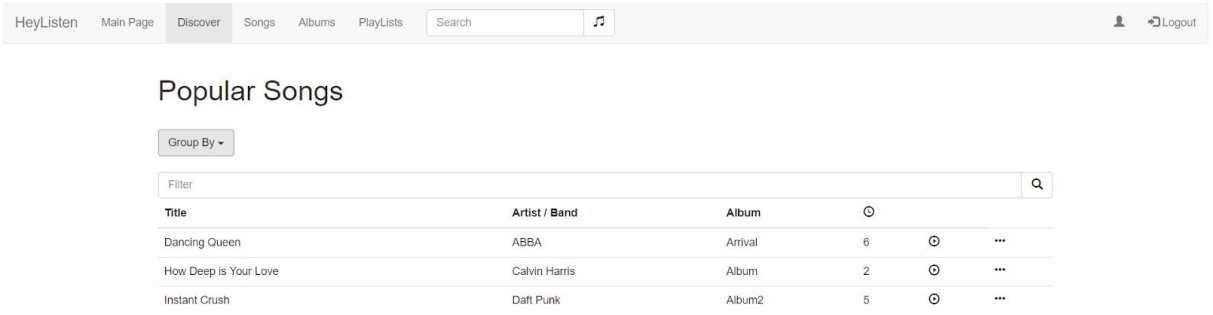
In this tab, user can view people she is following. She can unfollow them here.

7.10. Followers

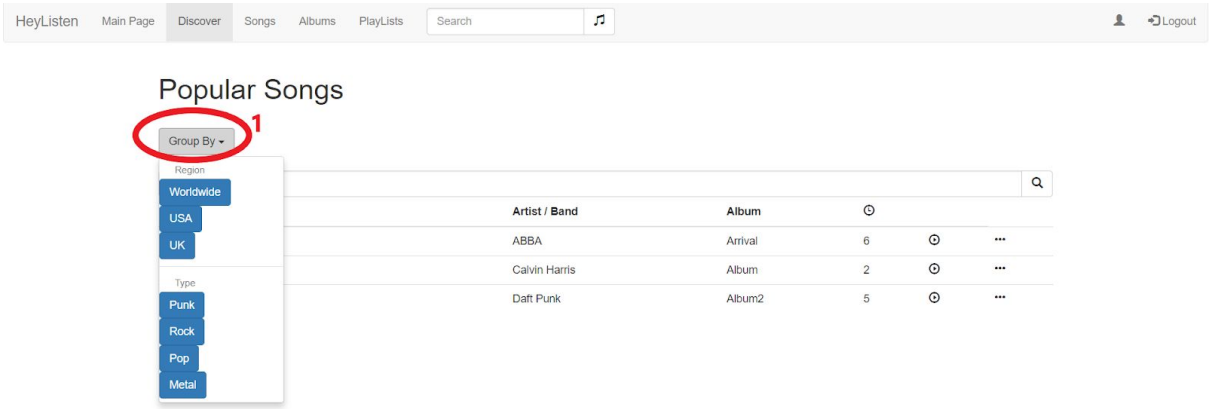


In this tab, user sees her followers. She can follow/unfollow them here.

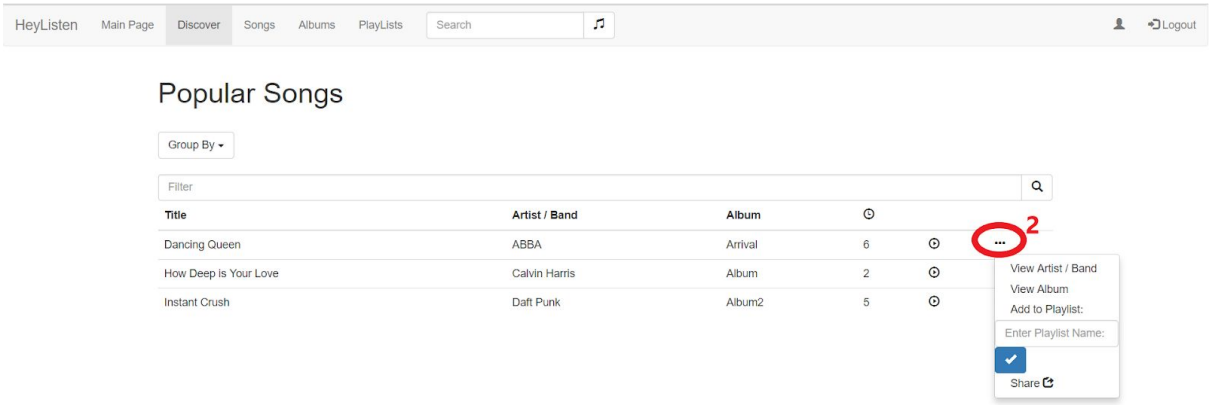
7.11. Discover



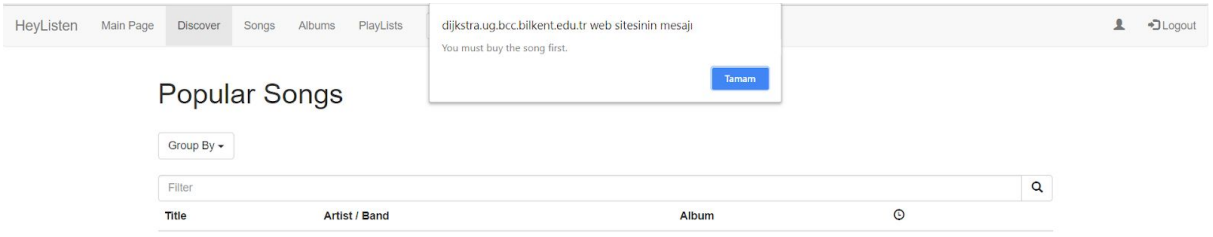
In discover page, user sees trending songs. She can filter the results based on type and region as follows:



Pressing to button 1 displays options to filter results. User can choose one of them to filter.

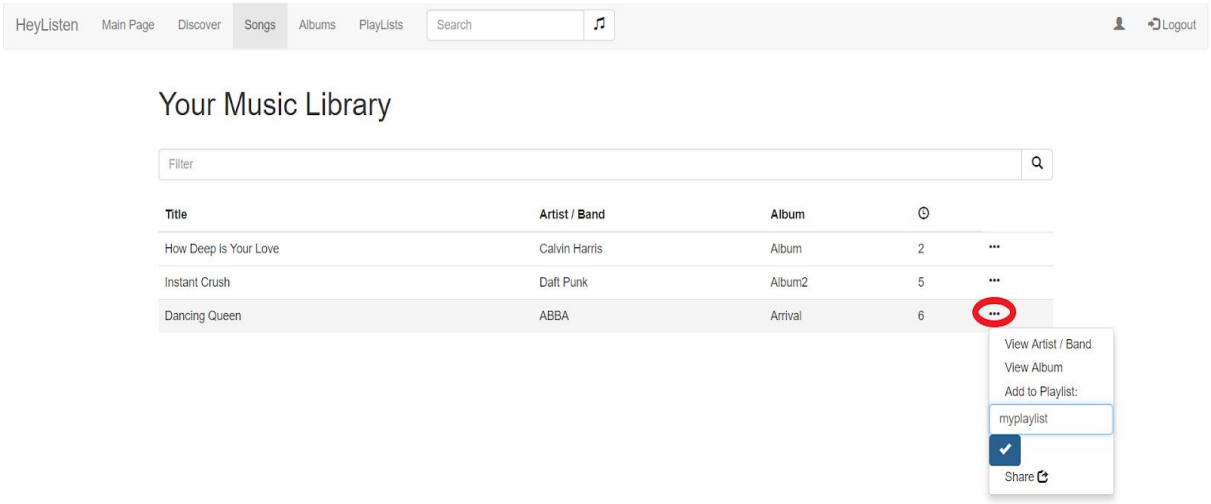


Using button 2, user has options to view artist's page, album's page and add song to one of her playlists.



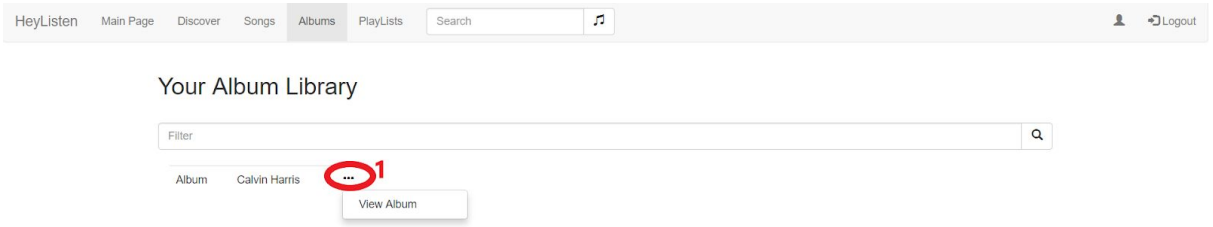
A user must buy a song before she adds it to a playlist. Otherwise, a warning message is displayed.

7.12. Songs



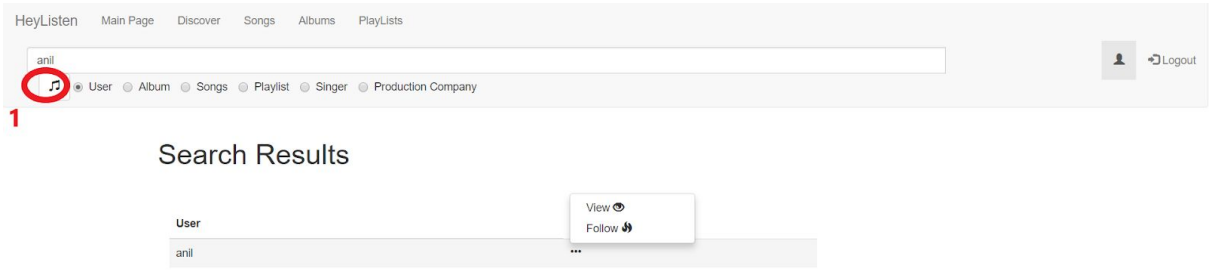
In songs page, user can see songs she has bought. Also, for each song, she has options to go to artist's page, album's page, or add the song to a playlist.

7.13. Albums

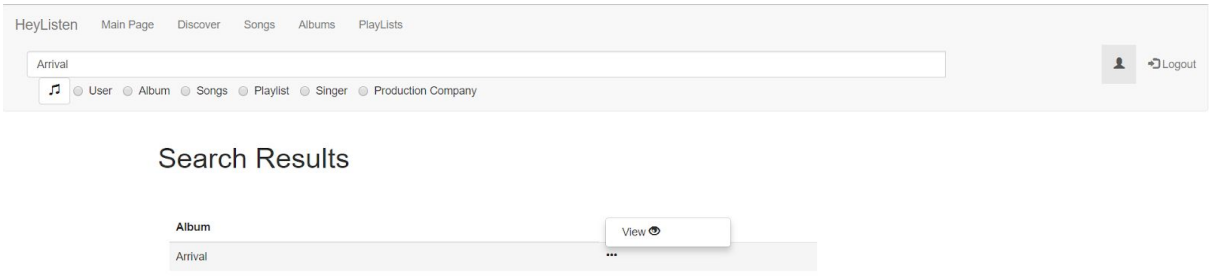


In albums page, user can see albums she has bought. An option to go to album's page is displayed when user presses button 1.

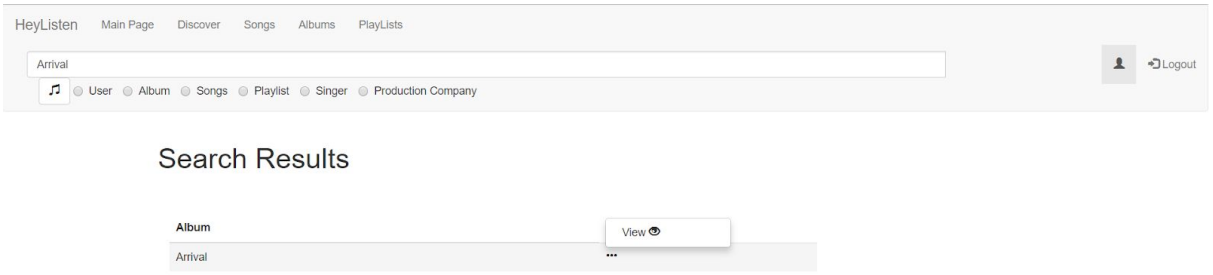
7.14. Search



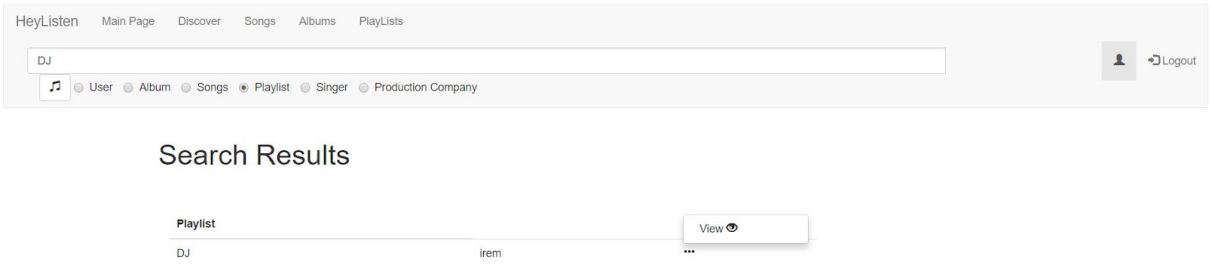
By using search bar and button 1, user can search for other users, albums, songs, playlists, singers or production companies. With search results, user can view a user's profile and follow/unfollow her.



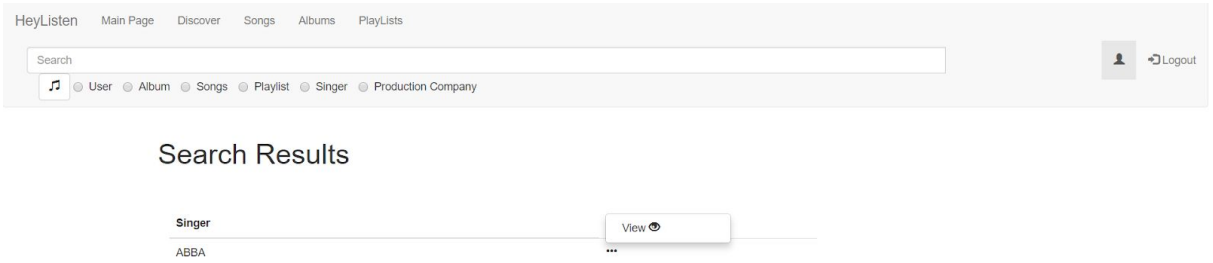
Album page can be viewed as in the above page.



Song page can be viewed.

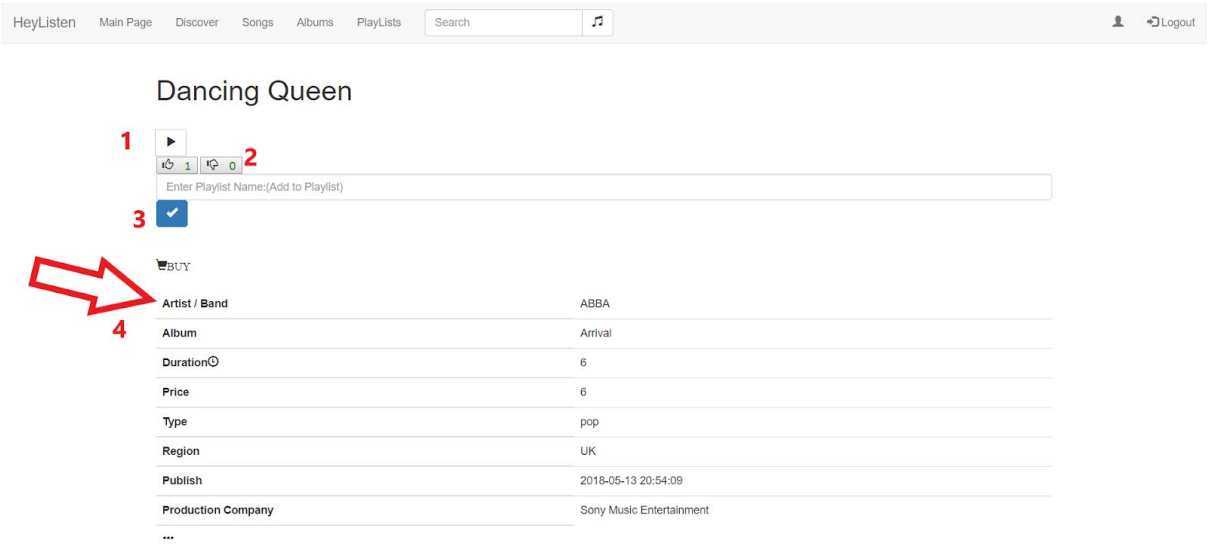


Playlist page can be viewed.

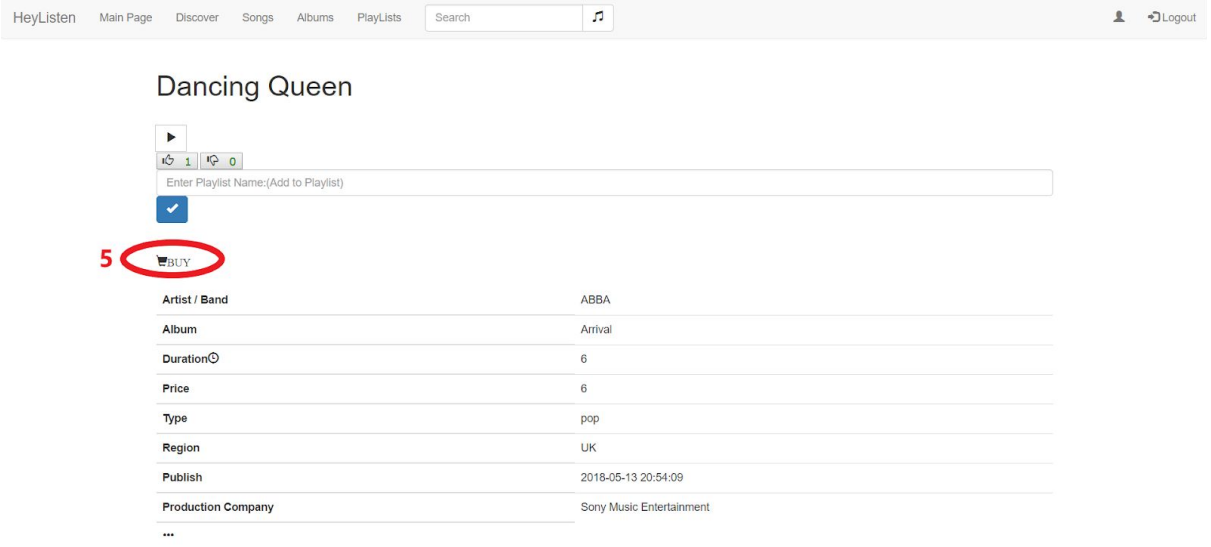


Singer page can be viewed.

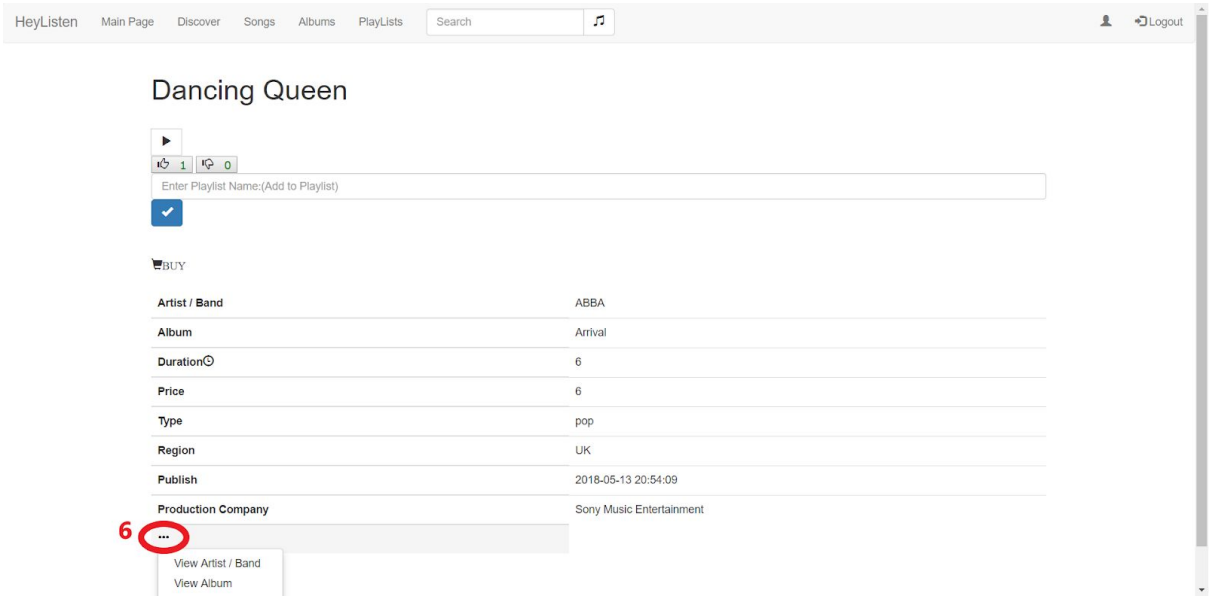
7.15. Song Page



In the song page, user can play the song (1), like/dislike it and see the number of likes/dislikes (2), add it to one of her playlists (3) and see information about the song like artist name, album name, duration, price, type, region, publish date and publisher company (4).

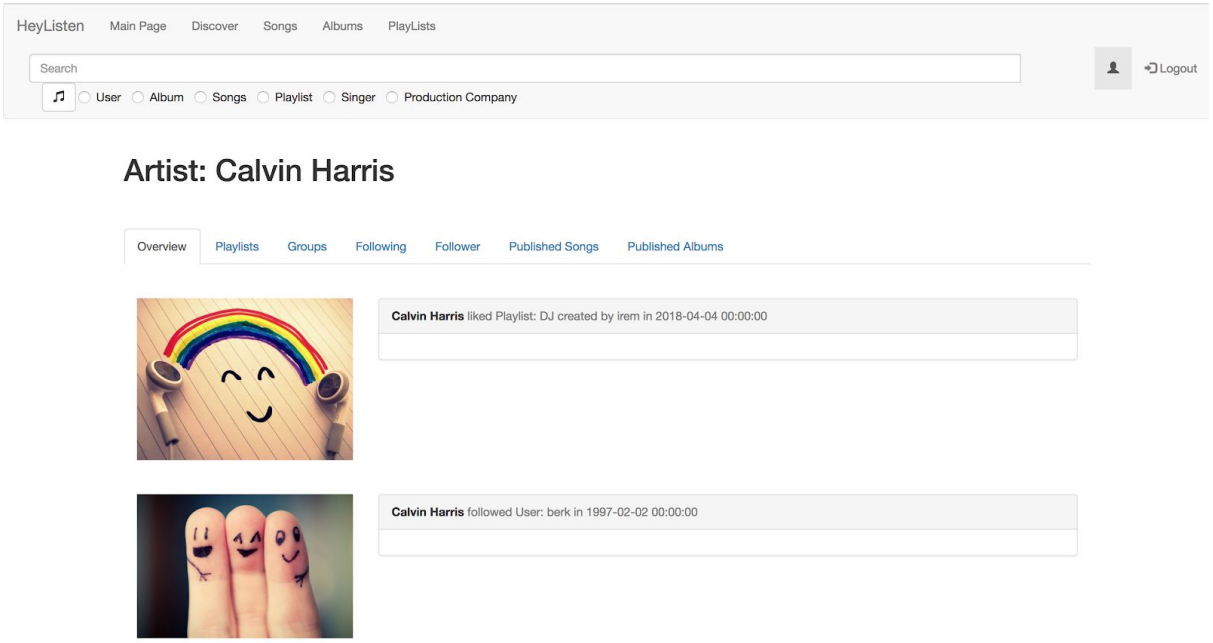


By button 5, user can buy the song. User must have enough budget to buy the song.



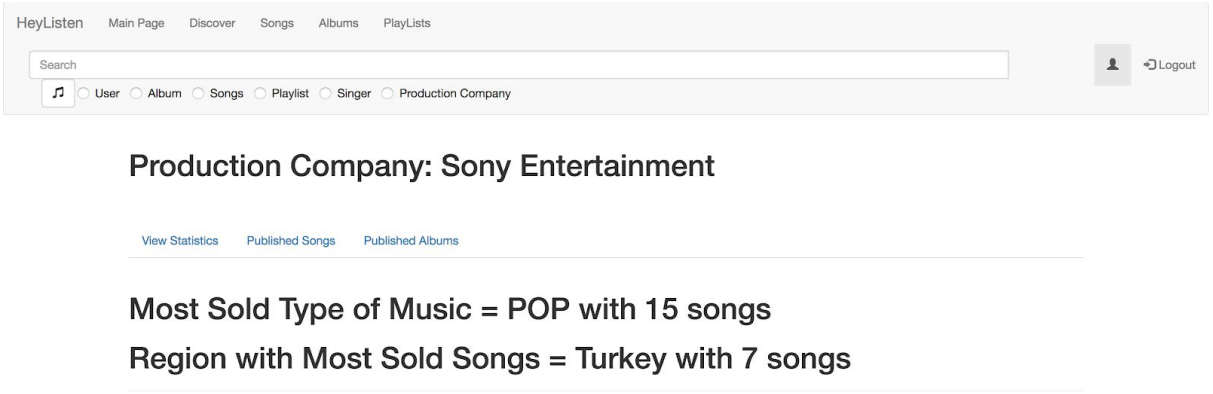
By button 6, user has options to go to artist's page or album's page.

7.16. Artist Page



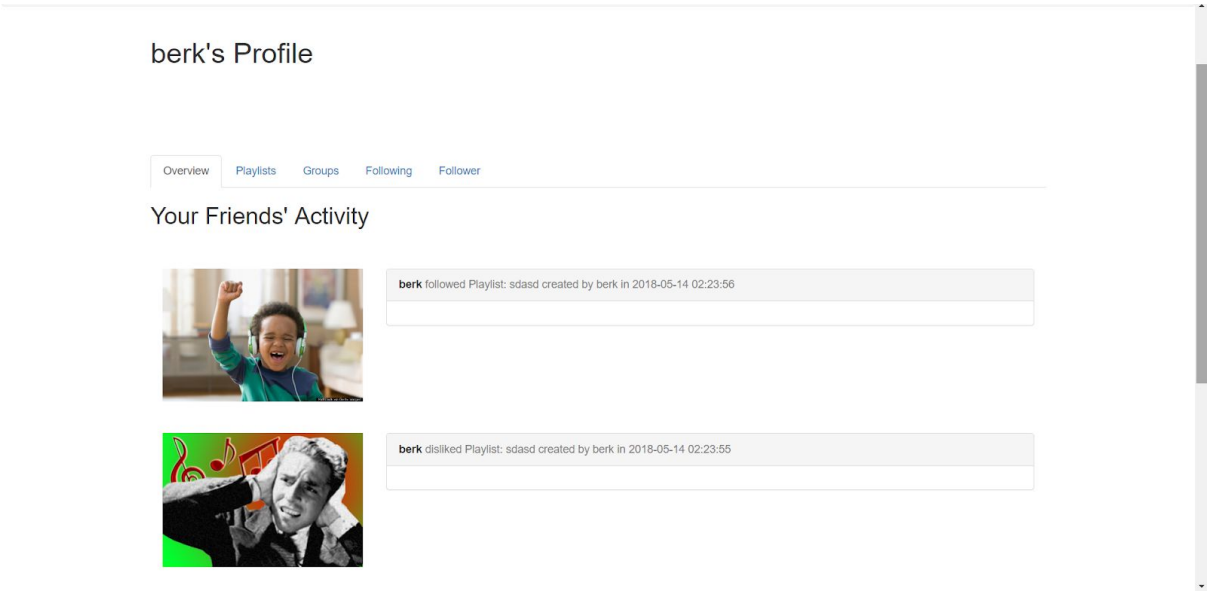
A user can view an artist's page and view her activities, playlists, groups, followings, followers and published songs and albums.

7.17.Production Company Page as User

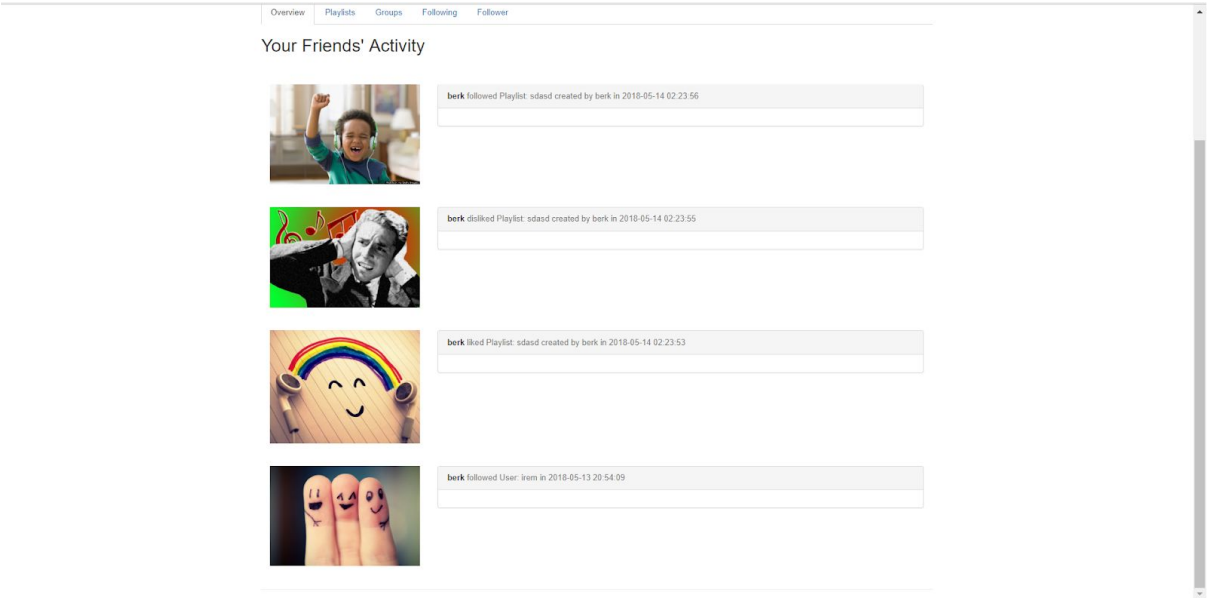


Production companies can view statistical data about the songs they have published. Moreover, they can view the list of songs and albums they have published.

7.18. Viewing Other Users' Profiles

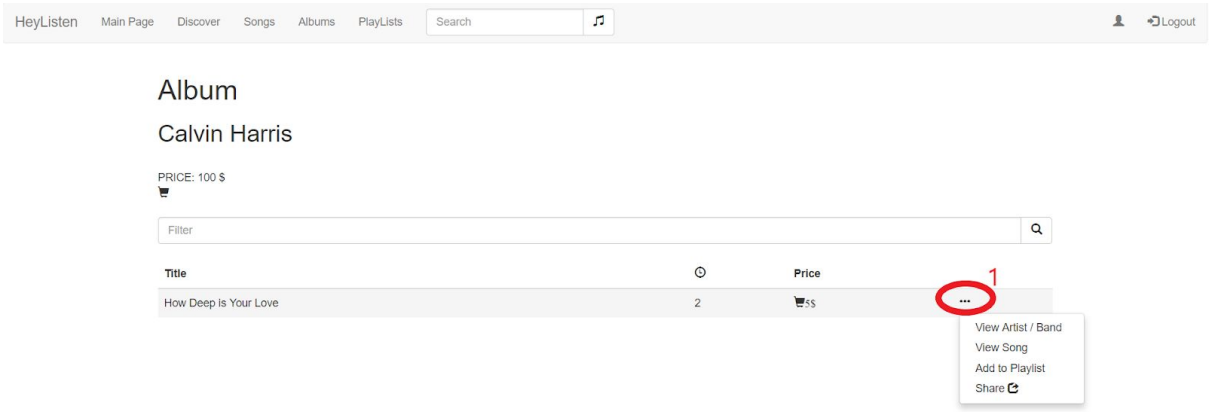


When user enters another user's profile, she can see her activities in overview tab and has options to navigate to other tabs to see her playlists, groups, followings and followers.

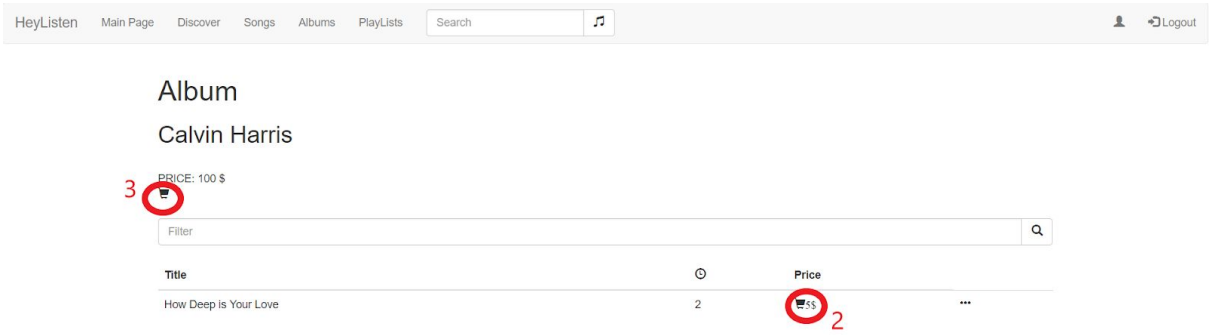


Activities include like, dislike, follow a playlist and follow a user.

7.19. Album Page

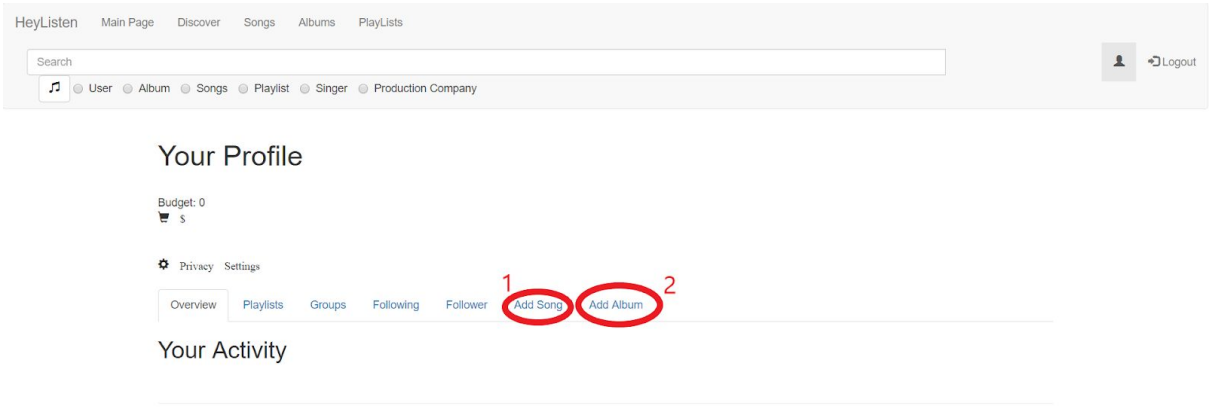


In album page, user sees the singer, price of the album and songs in the album. For each song, by using button 1, user can view artist's page, song's page and add it to a playlist.



User can buy a song from album by using button 2 and can buy whole album by button 3. To buy them, user must have enough budget.

7.20. Artist User Type



If user registers as “Musician” then she has options to add a song or album to the system by using buttons 1 and 2.