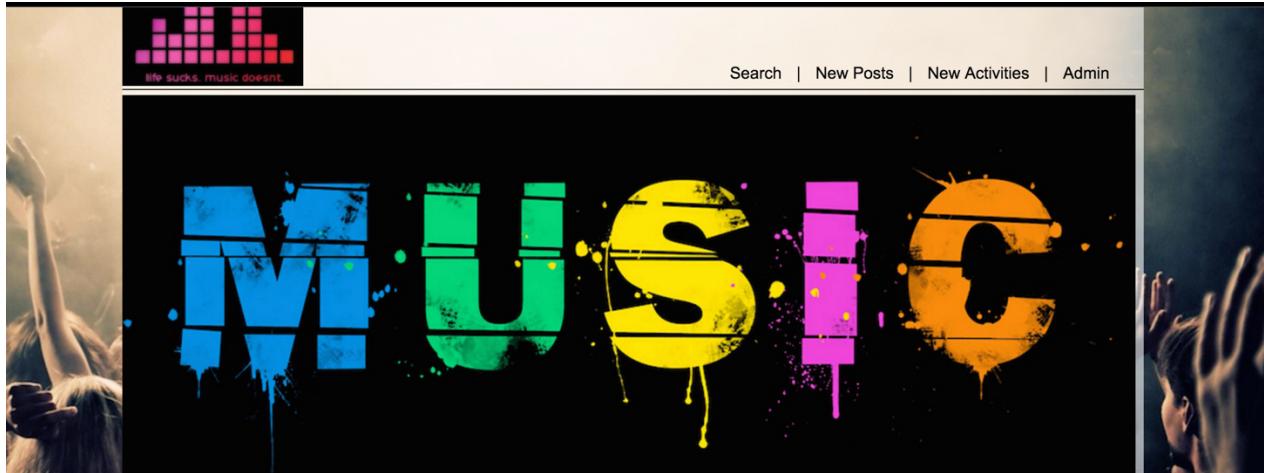
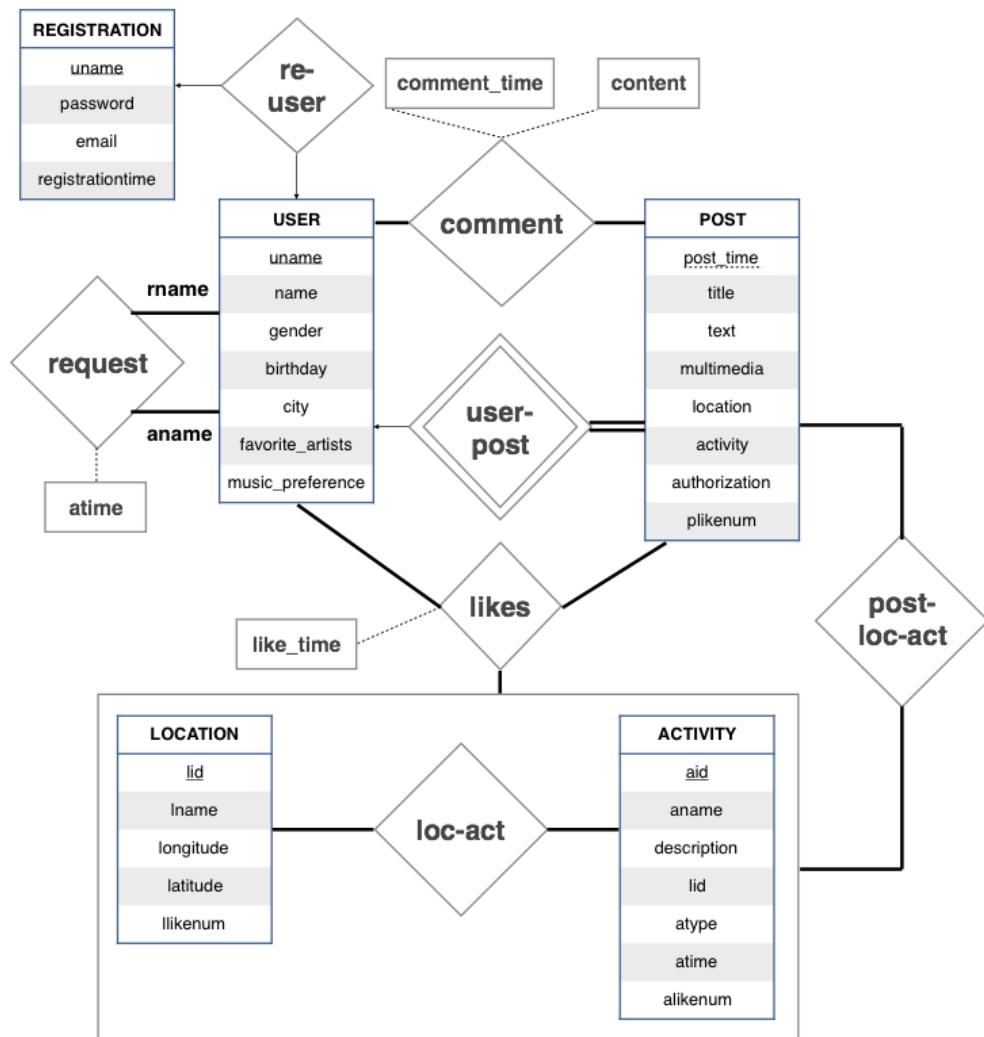


Welcome to Music!



(a). ER diagram and Description



Description: My application is a music fans social network.

Users are music fans of all kinds of music. They can have their own homepage and can manage their personal settings, such as create and edit profile, share personal posts with friends, create interesting activity to meet people off the line.

First music fans need to sign up with a unique username and email, along with their own passwords, stored in table REGISTRATION. After an account is created, users can edit their profile which contains basic personal information like real name, gender, birthday, home city, favorite artists and music preference. And of course users can update their profiles whenever they want to. The profiles are stored in table USER.

Then they can add friends and make connections on the network. First they need to know the username of the future friends. By searching specific username, they can get their profile and public posts before they send out a friend. After the request is accepted, they can officially be friends. The friend requests are stored in table REQUEST. The default value of accept time “atime” is null. After the request is accepted, it will be updated automatically into current timestamp. The FRIENDSHIP is a view of REQUEST, where the value of “atime” is not null.

They also can post about their favorite bands, music halls, news about concerts, photos with their idols, etc. All the posts are stored in table POST, and each post is identified by the a automatically increase ID number. To post, a title and a text are needed. Other information like photos, location and activity information are optional. However as long as a location is filled in, an embedded google map will be shown blow the post. And the default value of “authorization” is 0, which means that this post can be seen by anyone. And 1 means that this post can only be seen by friends. 2 means that this post can be seen by FOFs. Users can make a choice when they post.

All the posts can be commented by the users who can see it. All the comments are stored in COMMENT. And each comment is identified by the combination of the username who comment, the post ID of the post, and the comment time.

Besides comment, a post can also be liked or disliked by other music fans. There are attributes in tables POST, which is “plikenumber”, the number of likes. Every time a like is made, the system will insert a row into table LIKES, which stores all the likes. And at the same time, the corresponding

“plikenum” will be updated. The default values is 0. Each like is identified by the username and like time. And the DISLIKE is in the same procedure.

Table ACTIVITY stores all the activities which are identified by their own activity ID “aid”. And each activity has a corresponding location and activity type “atype”. The value of “atype” is “0” for “other” kinds of activities, “1” for “concert”, “2” for “party” and “3” for “music festival”.

(b). Convert the ER diagram into relational schemas.

REGISTRATION (uname, password, email, registrationtime)

USER (uname, name, gender, birthday, city, favorite_artists, music_preference)

foreign key (uname) references uname in REGISTRATION

REQUEST (rname, aname, atime)

foreign key (rname) & (aname) reference uname in REGISTRATION

POST (pid, pname, post_time, title, content, multimedia, location, authorization, plikenum)

foreign key (pname) references uname in REGISTRATION

COMMENT (cname, pid, comment_time, content)

foreign key (cname) references uname in REGISTRATION

foreign key (pid) reference pid in POST

LIKES (uname, like_time, pid)

foreign key (uname) references uname in REGISTRATION

DLIKES (uname, dlike_time, pid)

foreign key (uname) references uname in REGISTRATION

ACTIVITY (aid, aname, description, location, atype, atime, alikenum)

Create tables:

create table REGISTRATION (uname **varchar**(20) **primary key**,
password **varchar**(20), **not null** email **varchar**(50), **unique & not null**

registrationtime **timestamp & not null**)

create table USER (uname **varchar(20) primary key**, name **varchar(20) not null**, gender **varchar(8)**, birthday **date**, city **varchar(20)**, favorite_artists **text**, music_preference **text**)

foreign key (uname) **references** uname in REGISTRATION

create table REQUEST (rname **varchar(20) primary key**, aname **varchar(20) primary key**, atime **timestamp**)

foreign key (rname, aname) **references** uname in REGISTRATION

create table POST (pid **int(10) primary key**, pname **varchar(20)**, post_time **timestamp**, title **varchar(50) & not null**, content **text & not null**, multimedia **longblob**, location **varchar(40)**, authorization **int(1) & not null**, plikenum **int(10) & not null**)

foreign key (pname) **references** uname in REGISTRATION

default value of authorization is 0, which means that this post can be seen by anyone. And 1 means that this post can only be seen by friends. 2 means that this post can be seen by FOFs.

default value of plikenum is 0

create table COMMENT (cname **varchar(20) primary key**, pid **int(10) primary key**, comment_time **timestamp primary key**, content **text & not null**)

foreign key (cname) **references** uname in REGISTRATION **foreign key** (pname, post_time) **references** (pname, post_time) in POST

create table LIKES (uname **varchar(20) primary key**, like_time **timestamp primary key**, pid **varchar(20)**)

create table DLIKES (uname **varchar(20) primary key**, dlike_time **timestamp primary key**, pid **varchar(20)**)

foreign key (uname) **references** uname in REGISTRATION

foreign key (pid) **references** (pid) in POST

```
create table ACTIVITY (aid int(10) primary key, aname varchar(20) &  
not null, description text& not null, location varchar(40) & not null,  
atype int(1) & not null, atime date & not null)
```

(c). Stored Procedures

The number of someone's posts:

```
create procedure postnum(INOUT username VARCHAR(20), OUT  
postnum INT)
```

```
select COUNT(*) into postnum from POST where pname = username
```

Call for the procedure: **SET** @p0 = 'Rockfan' **CALL** postnum(@p0,
@p1) **select** @p0 **AS** username, @p1 **AS** postnum

The number of someone's friends:

```
create procedure friendsnum(INOUT username VARCHAR(20), OUT  
friendsnum INT)
```

```
select COUNT(*) into friendsnum from friendship where users =  
username
```

Call for the procedure: **SET** @p0 = 'Folks' **CALL** friendsnum(@p0,
@p1) **select** @p0 **AS** username, @p1 **AS** friendsnum

The number of someone's FOFs:

```
create procedure FOFsnum(INOUT username VARCHAR(20), OUT  
FOFsnum INT)
```

```
select COUNT(*) into FOFsnum from (select distinct second.friends  
from (friendship AS fst) join (friendship AS second) on (fst.friends =  
second.users)  
where fst.users = username AND second.friends <> username) AS FOF
```

Call for the procedure: **SET** @p0 = 'Folks' **CALL** FOFsnum(@p0,
@p1) **select** @p0 **AS** username, @p1 **AS** FOFsnum

Note: more stored procedures will be added when they are needed in the
second part.

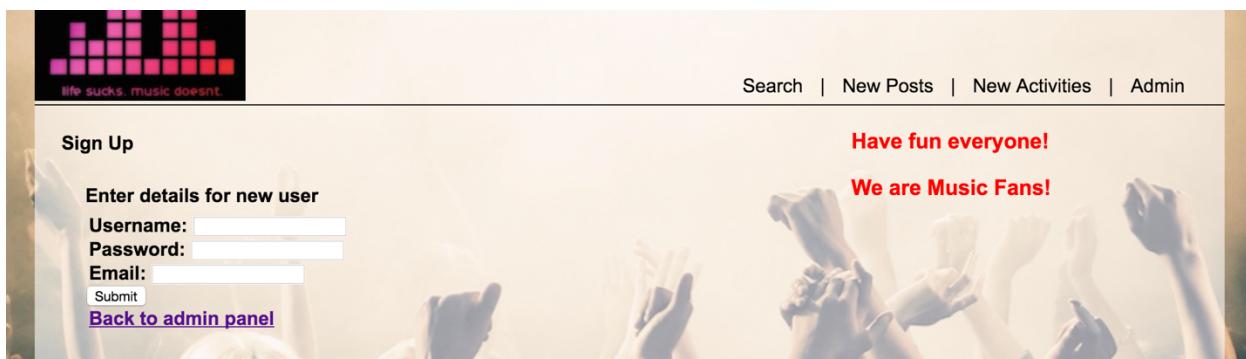
(d). Function introductions

1. Create a new account

Sign up:

**INSERT INTO REGISTRATION (uname, password, email, registrationtime)
VALUES**

(' Musicbuffs', musicbuffs, musicbuffs@musicfans.com',
CURRENT_TIMESTAMP AMP)



Log in page:



Create and edit the profile:

**INSERT INTO USER (uname, name, gender, birthday, city, favorite_artists,
music_preference)
VALUES**

(' Musicbuffs', 'Eric', 'Male','1992-11-03', 'New York', Adam Levine', 'Pop Rock')

The screenshot shows the Musicbuffs homepage. At the top left is a logo with a pink and red pixelated graphic and the text "life sucks. music doesn't.". At the top right are links for "Search", "New Posts", "New Activities", and "Admin". The main content area has a background image of a crowd with their hands raised. On the left, a message says "Hi, Musicbuffs. Your profile is quite empty. Edit it now to let your friends know more about you! [Back to my homepage](#) | [Edit profile](#)". On the right, there are two red messages: "Have fun everyone!" and "We are Music Fans!". A form titled "Enter details for your profile" is displayed, containing fields for Name (Eric), Gender (Male), Birthday (11/03/1992), City (New York), Favorite artists (Adam Levine), and Music preference (Pop Rock). A "Submit" button and a link "Back to my profile page" are also present.

Personal homepage:

The screenshot shows the personal homepage for a user named "Eric". At the top left is a logo with a pink and red pixelated graphic and the text "life sucks. music doesn't.". At the top right are links for "Search", "New Posts", "New Activities", and "Admin". The main content area has a background image of a crowd with their hands raised. On the left, a message says "Hi! Musicbuffs. [My Friends \(0\)](#) | [New Friend Requests \(0\)](#) | [View profile](#) | [Logout](#)". Below it is a "New Post" section with fields for "Title" (empty), "Content" (empty text area), "Photos" (choose file button, "No file chosen"), "Location" (empty input), "Authorization" (dropdown set to "Everyone can see"), and a "Post" button. A link "Go to Postwall" is also present. On the right, there are two red messages: "Have fun everyone!" and "We are Music Fans!". A "My Posts:" section below says "Sorry, you have no post here yet."

2. Make connection with other music fans

Add a new friend:

```
INSERT INTO REQUEST (rname, aname, atime) VALUES ('Musicbuffs', 'Rocker', NULL)
```



Search | New Posts | New Activities | Admin

Hi! Musicbuffs.
Sorry, you have no friend here yet.
[Find New Friends](#)
[Return to my homepage](#)

Have fun everyone!
We are Music Fans!

Search results:

Rocker

Name: Dean
Gender: Male
Birthday: 1989-11-10
City: LA
Favorite artists: Maroon 5
Music preference: Rock

Sorry, this user has no post here yet.
[Send Friends Request](#)

[Find New Friends](#)
[Return to my homepage](#)

Have fun everyone!
We are Music Fans!

Accept a new friend request:

UPDATE

REQUEST

SET

atime = CURRENT_TIMESTAMP

WHERE rname = 'Musicbuffs' AND fname = 'Rocker'



Search | New Posts | New Activities | Admin

Hi! Rocker.
[My Friends \(0\)](#) | [New Friend Requests \(1\)](#) | [View profile](#) | [Logout](#)

Have fun everyone!
We are Music Fans!

Here are your new friend request(s) from:

Musicbuffs | [Accept](#) | [Reject](#)

Name: Eric
Gender: Male
Birthday: 1992-11-03
City: New York
Favorite artists: Adam Levine
Music preference: Pop Rock

Post:
Sorry, this user has no post here yet.

[Return to my homepage](#)

Friendships:

List someone's all current friends:

CREATE VIEW friendship **AS**

select rname **AS** users, aname **AS** friends, atime **AS** time **from**
Request **where** (atime **is not null**) **union**

select aname **AS** users, rname **AS** friends, atime **AS** time **from**
Request **where** (atime **is not null**)

select users, friends **from** friendship **where** users = 'Rocker'

The screenshot shows a user profile for 'Rocker'. At the top, there is a black header bar with a logo consisting of a grid of pink and red squares and the text 'life sucks. music doesn't'. To the right of the logo are links for 'Search', 'New Posts', 'New Activities', and 'Admin'. Below the header, the profile information for 'Rocker' is displayed. On the left, it says 'Hi! Rocker.' and 'Friends:'. On the right, it says 'Have fun everyone!' and 'We are Music Fans!'. In the center, it shows a list of friends: 'Musicbuffs | since2016-05-09 04:49:41'. At the bottom, there are buttons for 'Find New Friends' and 'Return to my homepage'.

List someone's all current FOFs:

select distinct second.friends **from** (friendship **AS** fst) **join** (friendship **AS** second)

on (fst.friends = second.users) **where** fst.users = 'Folks' **AND**
second.friends <> 'Folks'

Assumption: we only consider one degree of FOFs. As we can see, user Folks only have two direct friends and four FOFs.

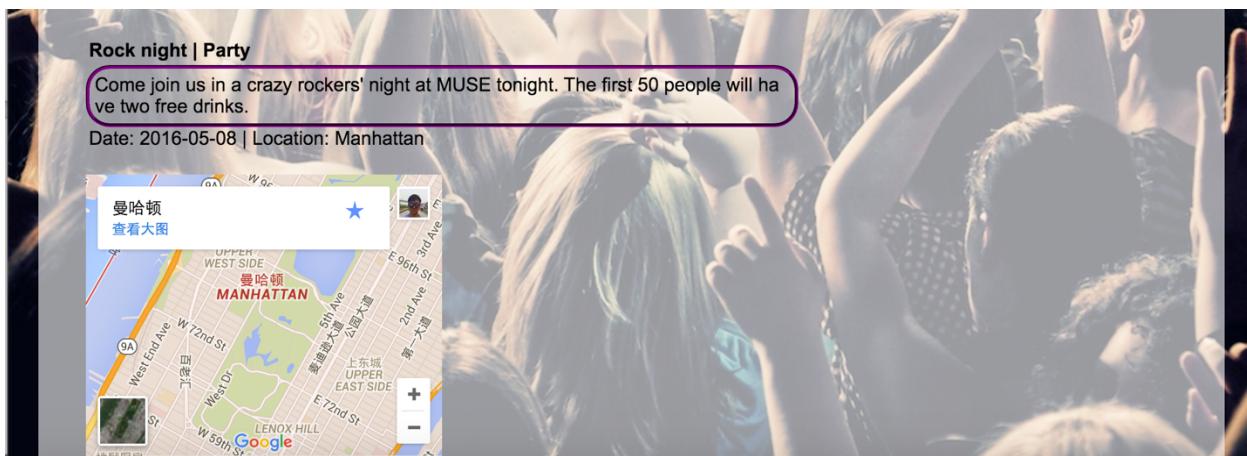
3. Create interesting activities

New activity:

The screenshot shows a website interface for creating a new activity. At the top, there's a logo with a pink pixelated graphic and the text "life sucks. music doesn't.". Below the logo, there are navigation links: "Search", "New Posts", "New Activities", and "Admin". The main content area has a background image of a crowd with their hands raised. On the left, there's a form for creating a new activity:

- Hi! Rocker.**
- Create Your Own New Activity**
- Activity Name:** Rock night
- Description:** Come join us in a crazy rockers' night at MUSE tonight. The first 50 people will have two free drinks.
- Location:** Manhattan
- Activity Type:** Party
- Date:** 05/08/2016
- Create** button

Below the form, a message says: "Sorry, there is no new activity right now."



4. Share posts

Post:

INSERT INTO POST (pname, post_time, title, content, multimedia, location, activity, authorization, plikenum)

VALUES

(Rocker, CURRENT_TIMESTAMP, 'Crazy night # Rock', 'Love the cocktail here! Great music, especially the Maroon 5 remix.

'Well done DJ Adam!', NULL, '13', '3', '0', '0')

life sucks. music doesn't.

Search | New Posts | New Activities | Admin

Hi! Rocker.
My Friends (1) | New Friend Requests (0) | View profile | Logout

New Post

Title: Crazy night # Rock

Content:
Love the cocktail here! Great music, especially the Maroon 5 remix.
Well done DJ Adam!

Photos: Choose File 142937vo3c...mxk35.jpg

Location: Manhattan

Authorization: Everyone can see

Post
Go to Postwall

My Posts:
Sorry, you have no post here yet.

Have fun everyone!
We are Music Fans!

Rocker: Crazy night # Rock

Love the cocktail here! Great music, especially the Maroon 5 remix. Well done DJ Adam!

Posted at: 2016-05-09 05:08:06 | Location: Manhattan

Like (0)
Dislike (0)

Comment

0 comment

0 comment

曼哈顿
查看大图

W 72nd St
West End Ave
W 59th St
Lenox Hill
E 72nd St
E 96th St
3rd Ave
2nd Ave
公园大道
第一大道
西区
东区
上东城
下东城
中城
西城
曼哈顿
MANHATTAN

Google 地图数据 ©2016 Google
地圖厨房 ELLIS KITCHEN 使用条款 报告地图错误

Comments:

INSERT INTO COMMENT (cname, pid, comment_time, content)

**VALUES ('Musicbuffs', 10, '2016-05-09 05:13:41', CURRENT_TIMESTAMP
AMP, 'Nice man! Did u get the two free drink? haha');**



Likes:

INSERT INTO LIKES (uname, like_time, pid)

VALUES ('Musicbuffs', CURRENT_TIMESTAMP, '10')

CREATE TRIGGER addlike AFTRE INSERT on LIKES for each row BEGIN UPDATE

LOCATION

SET

LOCATION.llikenum = LOCATION.llikenum + 1

WHERE

LOCATION.lid = new.lid;

UPDATE

ACTIVITY

SET

ACTIVITY.alikenum = ACTIVITY.alikenum + 1

WHERE

ACTIVITY.aid = new.aid;

UPDATE

POST

SET

POST.plikenum = POST.plikenum + 1

WHERE

POST.pname = new.pname AND POST.post_time = new.post_time;

END

CREATE TRIGGER cancellike **AFTRE INSERT** on LIKES for each row **BEGIN**

UPDATE

LOCATION

SET

LOCATION.llikenum = LOCATION.llikenum - 1

WHERE

LOCATION.lid = old.lid;

UPDATE

ACTIVITY

SET

ACTIVITY.alikenum = ACTIVITY.alikenum - 1

WHERE

ACTIVITY.aid = old.aid;

UPDATE

POST

SET

POST.plikenum = POST.plikenum - 1

WHERE

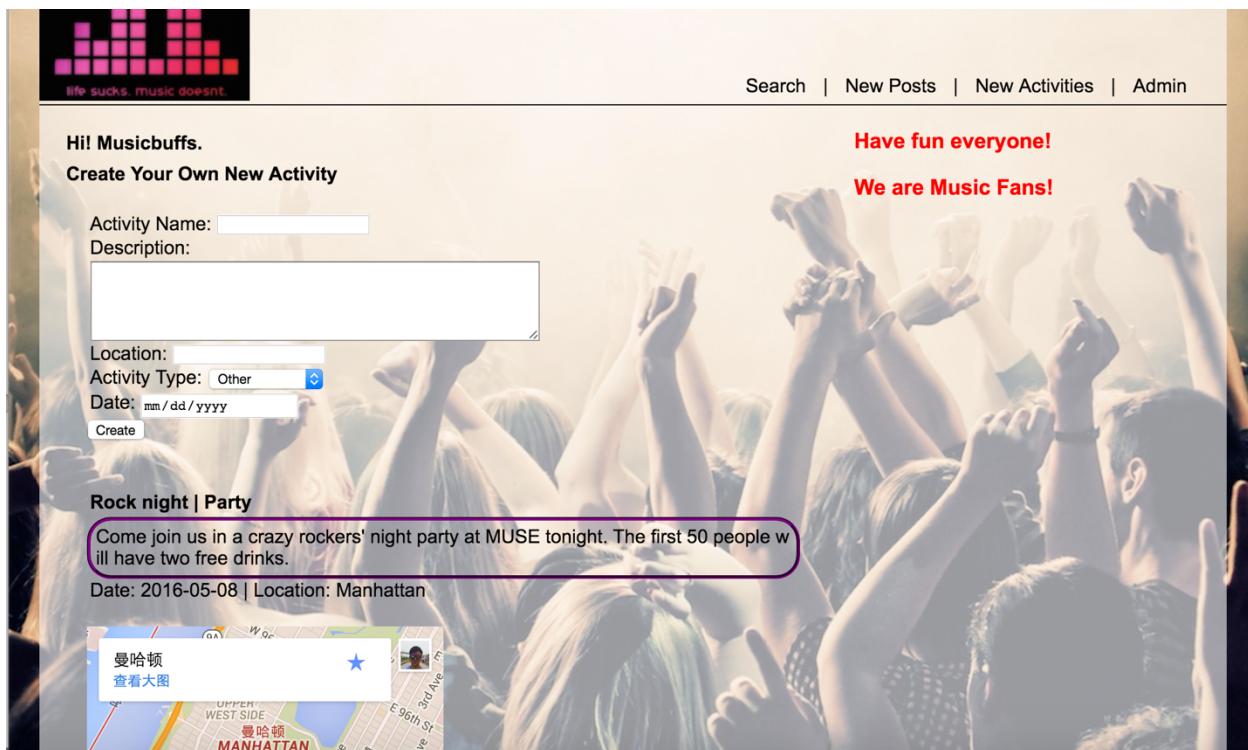
POST.pname = old.pname AND POST.post_time = old.post_time;

END

5. Browse/Search

We mainly have two “walls”, the “activity wall” and “post wall”.

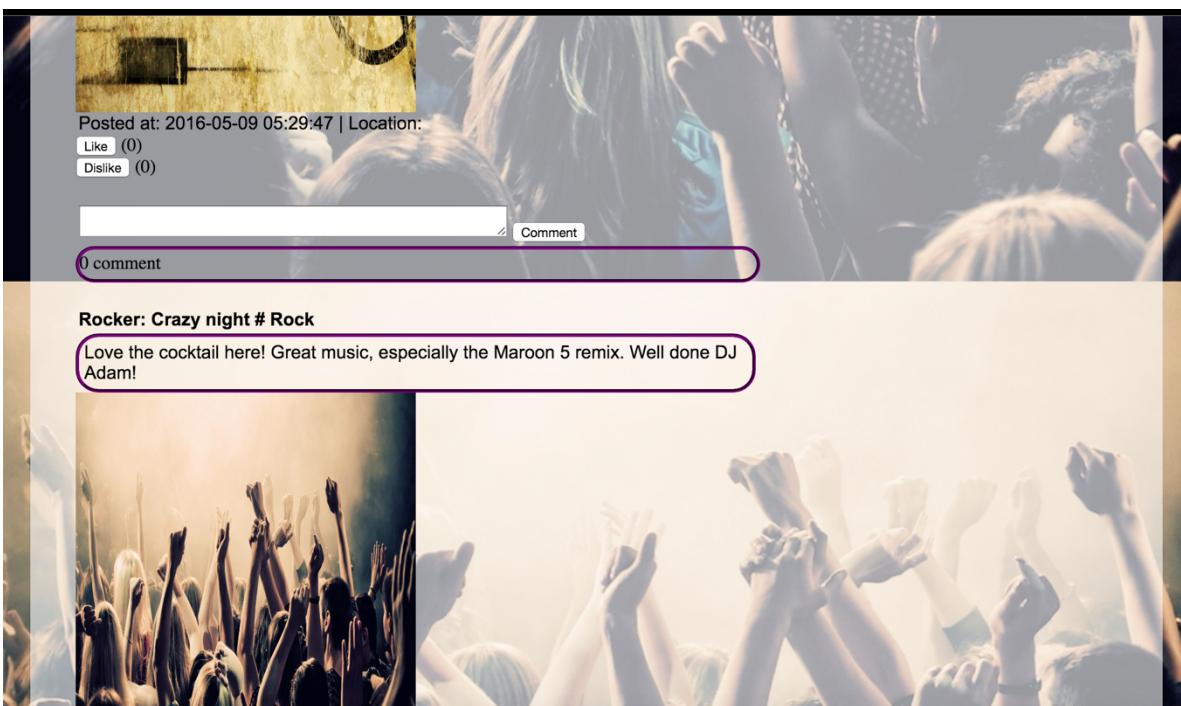
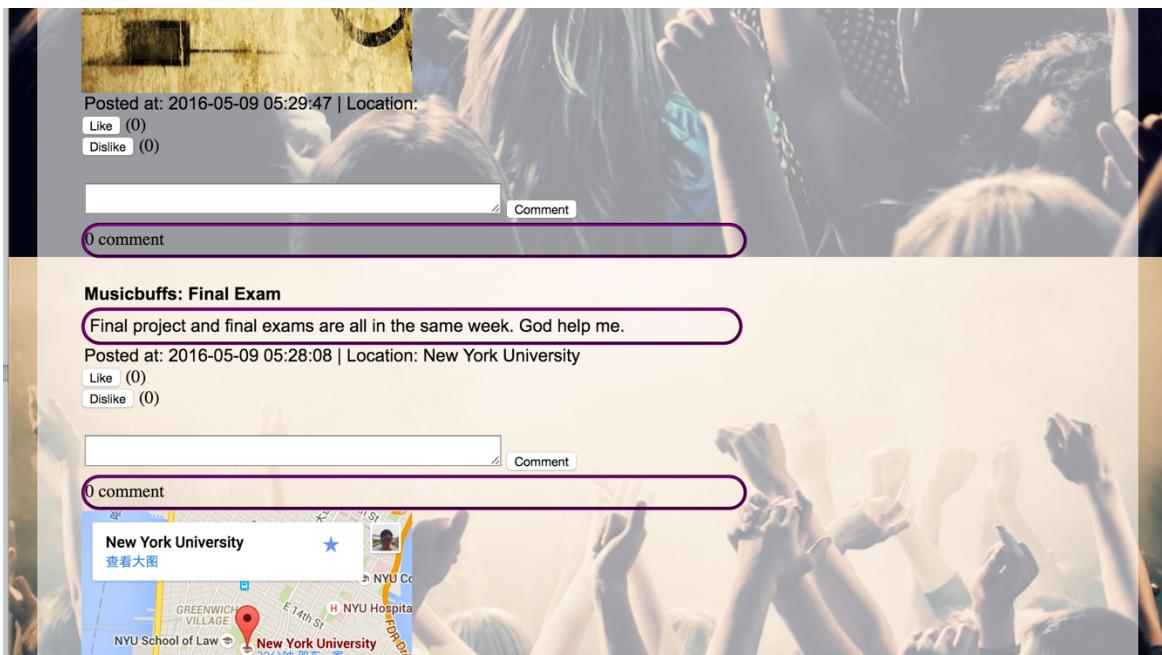
The “activity wall” shows all the activities created by music fans. And users create their new activity at that page as well.



The “post wall” shows all the public posts that can be seen by anyone and also users’ friends’ posts.

For example, musicbuffs and rocker are friends, so rocker can see more posts at the post wall, which are posted by musicbuffs and can be seen only by friends, while folks cannot, who is not musicbuffs’ friend.

So rocker can see the post about final exams posted by musicbuffs while folks cannot.



Search:

We also have a search function which can help the user to find users, posts, or activities they are interested in.



A screenshot of a search results page. The top section is identical to the previous one, featuring the logo and navigation bar. Below the search bar, there are links for "Continu search" and "Back to my home page". The main content area displays search results under sections: "Search results:", "Related Users:", and "Related Posts:". Under "Related Users:", it says "Sorry, there is no related user right now." Under "Related Posts:", it says "Sorry, there is no related post right now." Under "Related Activities:", there is a listing for "Rock night | Party" with a description: "Come join us in a crazy rockers' night party at MUSE tonight. The first 50 people will have two free drinks." This description is enclosed in a purple rounded rectangle. Below this, there is event information: "Date: 2016-05-08 | Location: Manhattan". To the right of the text, there is a small map of Manhattan with a pin pointing to the center labeled "曼哈顿" and "MANHATTAN". The map also shows street names like West End Ave, W 72nd St, and E 72nd St, along with Central Park and Lenox Hill.