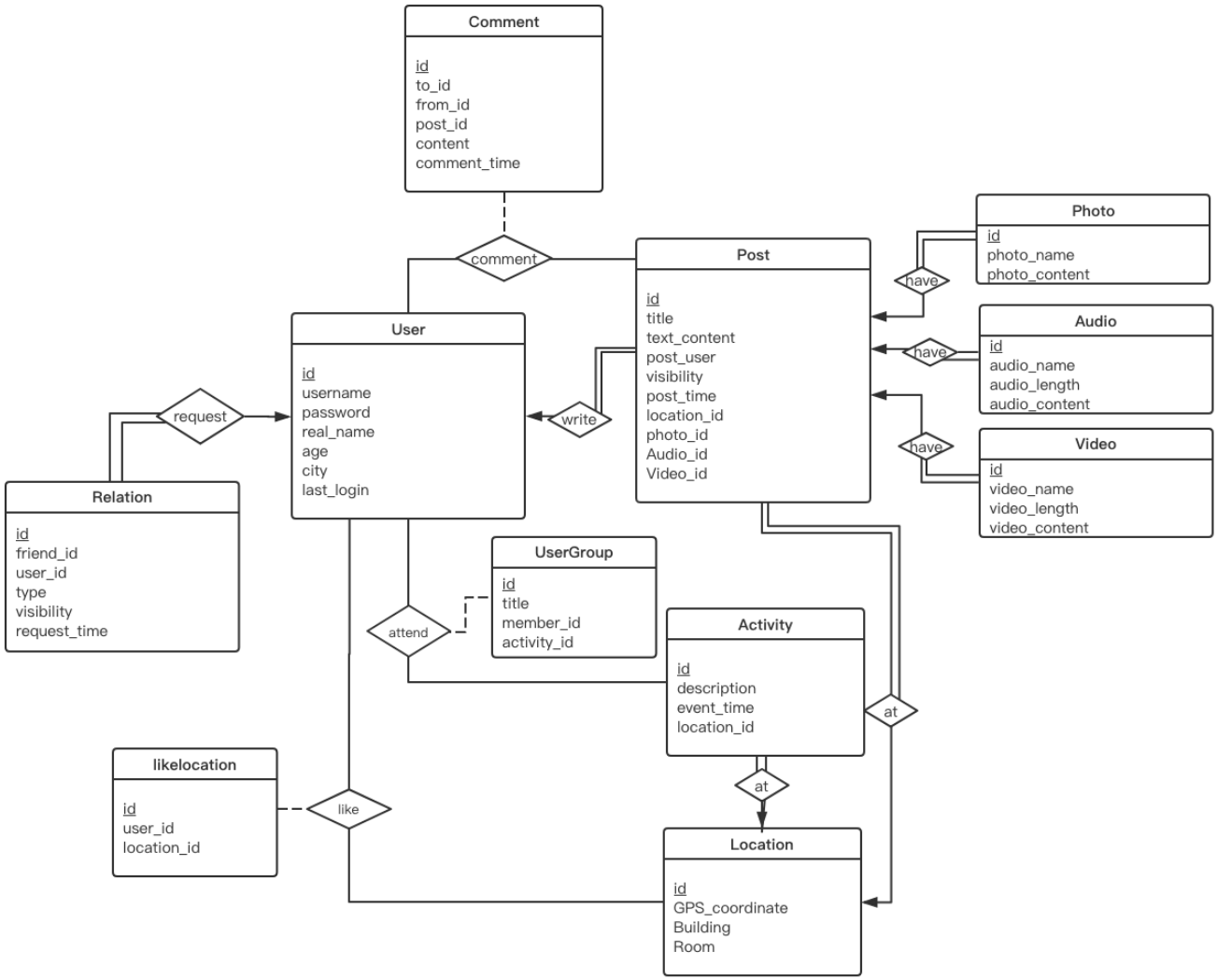
Project University students social network

Name: Xucheng Tang NetID: xt544

Part 1 Database Design

✦E-R diagram For University students social network:

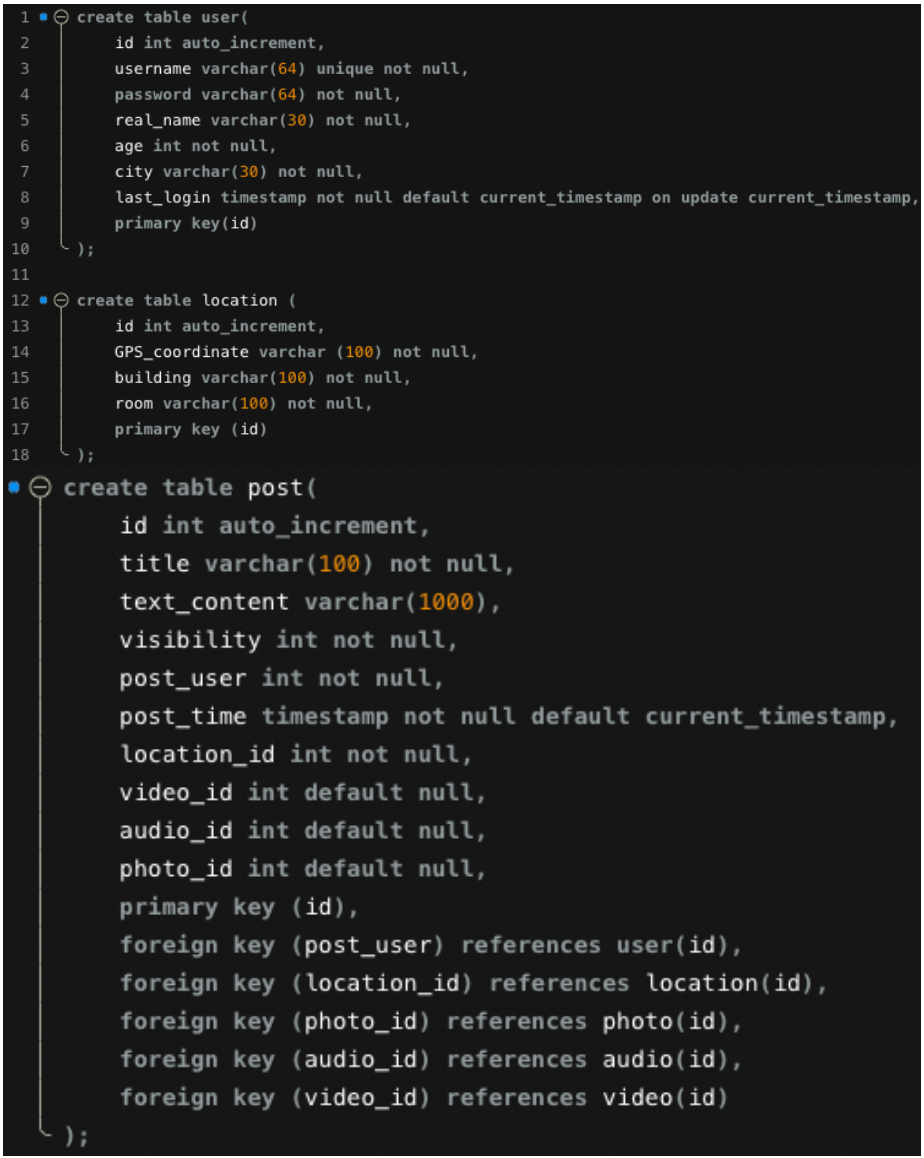


✦Schema For University students social network:

User (id, real_name, age, city, last_login)
Post (id, title, text_content, visibility, post_user, post_time, location_id, photo_id, video_id, audio_id)
Activity (id, location_id, description, event_time)
Comment (id, from_id, to_id, post_id, content, comment_time)
Location (id, GPS_coordinate, building, room)
UserGroup (id, title, activity_id, member_id)
Relation (id, user_id, friend_id, type, visibility, request_time)
Video (id, video_name, video_length, video_content)

Photo (id, photo_name, photo_content)
Audio (id, audio_name, audio_length, audio_content)
LikeLocation (id, user_id, location_id)
Post.post_user references user.id
Post.location_id references location.id
Post.video_id references video.id
Post.Audio_id references audio.id
Post.photo_id references photo.id
Activity.location_id references location.id
Comment.from_id references user.id
Comment.to_id references user.id
Comment.post_id references post.id
UserGroup.activity_id references activity.id
UserGroup.member_id references user.id
Relation.user_id references user.id
Relation.friend_id references user.id
LikeLocation.user_id references user.id
LikeLocation.location_id references location.id

- **The Database schema creating Process (DDL)**



```
create table user(
    id int auto_increment,
    username varchar(64) unique not null,
    password varchar(64) not null,
    real_name varchar(30) not null,
    age int not null,
    city varchar(30) not null,
    last_login timestamp not null default current_timestamp on update current_timestamp,
    primary key(id)
);

create table location (
    id int auto_increment,
    GPS_coordinate varchar (100) not null,
    building varchar(100) not null,
    room varchar(100) not null,
    primary key (id)
);
create table post(
    id int auto_increment,
    title varchar(100) not null,
    text_content varchar(1000),
    visibility int not null,
    post_user int not null,
    post_time timestamp not null default current_timestamp,
    location_id int not null,
    video_id int default null,
    audio_id int default null,
    photo_id int default null,
    primary key (id),
    foreign key (post_user) references user(id),
    foreign key (location_id) references location(id),
    foreign key (photo_id) references photo(id),
    foreign key (audio_id) references audio(id),
    foreign key (video_id) references video(id)
);
```

```sql
create table activity (
    id int auto_increment,
    location_id int not null,
    description varchar(500) not null,
    event_time Datetime not null,
    primary key (id),
    foreign key (location_id) references location(id)
);

create table comment (
    id int auto_increment,
    from_id int not null,
    to_id int not null,
    post_id int not null,
    content varchar(500) not null,
    comment_time timestamp not null default current_timestamp,
    primary key (id),
    foreign key (from_id) references user(id),
    foreign key (to_id) references user(id),
    foreign key (post_id) references post(id)
);
create table usergroup (
    id int auto_increment,
    title varchar(100) not null,
    member_id int not null,
    activity_id int not null,
    primary key (id),
    foreign key (member_id) references user(id),
    foreign key (activity_id) references activity(id)
);
create table relation (
    id int auto_increment,
    user_id int not null,
    friend_id int not null,
    status varchar(10) not null,
    visibility int not null,
    request_time timestamp not null default current_timestamp,
    primary key (id),
    foreign key (user_id) references user(id),
    foreign key (friend_id) references user(id)
);

create table video (
    id int auto_increment,
    video_name varchar(100) not null,
    video_length int not null,
    video_content longblob not null,

    primary key (id),

);

create table audio (
    id int auto_increment,
    audio_name varchar(100) not null,
    audio_length int not null,
    audio_content mediumblob not null,

    primary key (id),

);
```

```
97  create table photo (
98      id int auto_increment,
99      photo_name varchar(100) not null,
100     photo_content mediumblob not null,
101
102     primary key (id),
103
104 );
create table likelocation(
    id int auto_increment,
    user_id int not null,
    location_id int not null,
    primary key (id),
    foreign key (user_id) references user(id),
    foreign key (location_id) references location(id)
);
```

✦Some queries for the following tasks (DML)

1. Content Posting

- *User signs up (insert information about himself)*

  insert into user(username, password, real_name, age, city)
  value
  ('tom0125','12345', 'Tom Johnson', 24, 'New York'),
  ('amy001','23456','Amy Wong', 20,'Paris'),
  (‘Jack1', 'ac123', 'Jack Williams', 21, 'Beijing'),
  ('Ding', 'abc123', 'Ding Jie', 23, 'Shanghai'),
  ('jerry23','xyz123','Jerry Clark', 21, ‘Tianjin’);

- *User edits personal profile*

  update user set city = 'Nanjing', age = 22
  where id = 4;

- *Users post images (which included in a post)*

  insert into location(GPS_coordinate, building, room)
  value ('41°21\'N 2°12\'E', 'Watson building', ‘302'),
  ('39°24\'N 1°42\'E', ‘Metro-tech building', ‘501');
  insert into photo(photo_name, photo_content)
  value
  ('selfies', '/opt/pic/5.jpg'),
  ('information', '/opt/pic/1.jpg')
  insert into post(title, text_content, visibility, post_user, location_id, photo_id)
  value
  ('April 11th Diary', 'I post some selfies.',2, 2, 1, 1),
  ('Get Information', 'Look at this', 3, 1, 2, 2);

- *Users register to events*

  insert into location(GPS_coordinate, building, room)
  value ('47°21\'N 4°12\'E', ‘Game lab', '208');

  insert into activity(location_id, description, event_time)
  value
  (3, 'Internship fair', '2019-4-10 11:00:00'),
  (2, 'Union Party', '2019-4-4 20:30:00');

  insert into usergroup(title, member_id, activity_id)
  value ('internship group',2, 1),('party group',1, 2),('party group',2, 2);
  —This insertion means we create two activities, user 1 attends second activity, and user 2 attends activity 1 and 2.

2. Friendship

In my friendship design, I create a new table called relation. If one user request a friendship with other user, then it will insert a new row in the table, status will be ‘pending’. Then if the other user reply to accept this friendship, system will use a trigger to update the ‘pending’ status to ‘friend’ status and there will be two lines to indicate a single friendship. This may cause some redundant situations but it is safe to say this request-accept method works. Then I record the friendship visibility desire, if one of them have smaller visibility: All>FOF>DF.

- *User defines relationship*

```
insert into relation(user_id, friend_id, status, visibility)
value (1, 2, ‘pending', 2);
-- It means user 1 request friendship to user 2. The friendship visibility is FOF.
```

- *User confirms relationship*

```
insert into relation(user_id, friend_id, status, visibility)
value (2, 1, 'friend', 2);
update relation set status = 'friend', visibility = 2
where user_id = 1 and friend_id = 2;
—use stored procedure to finish this update.
```

- *To select DF of user 1.*

```
select friend_id
from relation
where user_id = 1 and status = ‘friend';
```

- *To select FOF of user 1.*

```
select friend_id
from relation
where user_id = 1 and status = 'friend'
union
select distinct friend_id
from relation
where friend_id <>1 and status = 'friend' and user_id in (select friend_id
                                                          from relation
                                                          where user_id = 1
                                                    and status = 'friend') ;
```

3. Browse/Search Queries

Due to our relation table setting, we can easily find out the friend of a user, just pick out user_id. It also makes FOF easier to deal with, we just need to find out friends of all the friends of the specific user. So we can query the database like below.

- *To view profile from DF of user 1.*

```
select id, username, real_name, age, city, last_login
from user
where id in (select friend_id
                    from relation
                    where status = 'friend' and user_id = 1)
order by last_login DESC;
```

- *To view text post from FOF of user 1.*

```
select post_user, title, text_content, post_time
from post
where post_user in (select friend_id
from relation
where user_id = 1 and status = 'friend'
union
select distinct friend_id
from relation
where friend_id <>1 and status = 'friend' and user_id in (select friend_id
                                                          from relation
                                                          where user_id = 1
and status = 'friend') )
order by post_time DESC;
```

- *To view multimedia post from DF of user 1.*

```
select post_user, title, text_content, post_time, photo_content, video_content,
audio_content, post_time, location_id
from post left outer join photo on photo.id = post.photo_id left outer join video on
video.id = post.video_id left outer join audio on audio.id = post.audio_id
where post_user in (select friend_id
from relation
where user_id = 1 and status = 'friend')
order by post_time desc;
```

- *To view multimedia post from FOF of user 1.*

```
select post_user, title, text_content, post_time, photo_content, video_content,
audio_content, post_time, location_id
from post left outer join photo on photo.id = post.photo_id left outer join video on
video.id = post.video_id left outer join audio on audio.id = post.audio_id
where post_user in (select friend_id
from relation
where user_id = 1 and status = 'friend'
union
select distinct friend_id
from relation
where friend_id <>1 and status = 'friend' and user_id in (select friend_id
                                                          from relation
                                                          where user_id = 1 and
status = 'friend') )
order by post_time desc;
```

- *To view multimedia post from all users.*

```
select post_user, title, text_content, post_time, photo_content, video_content,
audio_content, post_time, location_id
from post left outer join photo on photo.id = post.photo_id left outer join video
on video.id = post.video_id left outer join audio on audio.id = post.audio_id
where post_user in (select id from user)
order by post_time desc;
```

- *To view multimedia post from FOF of user 2 which contains words “ice hockey".*

```
select post_user, title, text_content, post_time, photo_content, video_content,
audio_content, post_time, location_id
from post left outer join photo on photo.id = post.photo_id left outer join video on
video.id = post.video_id left outer join audio on audio.id = post.audio_id
where text_content like '%ice hockey%' or title like '%ice hockey%'and post_user in
(select friend_id
from relation
where user_id = 2 and status = 'friend'
union
select distinct friend_id
from relation
where friend_id <>2 and status = 'friend' and user_id in (select friend_id
                                                          from relation
                                                          where user_id = 2 and status =
'friend') )
order by post_time desc;
```

- *To view multimedia post from DF of user 1 from last week.*

```
select post_user, title, text_content, post_time, photo_content, video_content,
audio_content, post_time, location_id
from post left outer join photo on photo.id = post.photo_id left outer join video on
video.id = post.video_id left outer join audio on audio.id = post.audio_id
where post_time>unix_timestamp(NOW())-3600*24*7 and post_user in (select friend_id
from relation
where user_id = 1 and status = 'friend')
order by post_time desc;
```

- *To view all liked location from DF of user 1.*

```
select location.id, location.GPS_coordinate, location.building, location.room
from location join likelocation on location.id = likelocation.location_id
where user_id in (select friend_id
from relation
where user_id = 1 and status = 'friend');
```

✦Sample Data tables for this network

User

id	username	password	real_name	age	city	last_login
1	tom0125	12345	Tom Johnson	24	New York	2019-04-13 11:13:50
2	amy001	23456	Amy Wong	20	Paris	2019-04-13 11:13:50
3	Jackac	ac123	Jack Williams	21	Beijing	2019-04-13 11:13:50
4	Ding	abc123	Ding Jie	22	Nanjing	2019-04-13 11:19:19
5	jerry23	xyz123	Jerry Clark	21	Tianjin	2019-04-13 11:14:46

Post

id	title	text_content	visibility	post_user	post_time	location_id	video_id	photo_id	audio_id
1	April 11th Diary	I post some selfies.	2	2	2019-04-13 11:39:23	1	NULL	1	NULL
2	Get Information	Look at this	3	1	2019-04-13 11:39:23	2	NULL	2	NULL
3	personal diary	User 3 diary for today	2	3	2019-04-13 17:27:36	1	NULL	NULL	NULL
4	title	awesome day	1	4	2019-04-13 17:27:36	2	NULL	NULL	1
5	April 14th Diary	I got some videos.	2	2	2019-04-13 18:06:48	3	1	NULL	NULL
6	Some helpful things	1.Google, 2.Facebook	3	3	2019-04-13 18:06:48	2	NULL	NULL	NULL
7	Does someone want to play ice hockey?	I have some friends.	2	4	2019-04-13 21:01:06	5	NULL	NULL	NULL
8	Ice hockey playing	Play with us!	2	3	2019-04-13 21:02:01	5	2	NULL	NULL

Comment

	id	from_id	to_id	post_id	content	comment_time
▶	1	3	1	2	That's great!	2019-04-13 17:34:02
	2	2	4	4	You look happy today	2019-04-13 17:34:02

Audio

	id	audio_name	audio_length	audio_content
▶	1	1.mp3	40	BLOB

Video

	id	video_name	video_length	video_content
▶	1	1.mp4	140	BLOB
	2	ice ball	120	BLOB

Photo

	id	photo_name	photo_content
▶	1	selfies	BLOB
	2	information	BLOB

Activity

	id	location_id	description	event_time
	1	3	Internship fair	2019-04-10 11:00:00
	2	2	Union Party	2019-04-04 20:30:00

Relation

	id	user_id	friend_id	status	visibility	request_time
▶	1	1	2	friend	2	2019-04-13 15:59:36
	2	2	1	friend	2	2019-04-13 16:52:11
	3	1	3	friend	1	2019-04-13 16:57:45
	4	1	4	pending	3	2019-04-13 16:57:45
	5	3	1	friend	1	2019-04-13 16:57:45
	6	2	4	friend	2	2019-04-13 16:57:45
	7	4	2	friend	2	2019-04-13 16:57:45

Location

	id	GPS_coordinate	building	room
▶	1	41°21′N 2°12′E	Watson building	302
	2	39°24′N 1°42′E	Metrotech building	501
	3	47°21′N 4°12′E	Game lab	208
	4	47°21′N 4°12′E	Game lab	208
	5	44°10′N 5°20′E	Tandon GYM	null

LikeLocation

	id	user_id	location_id
▶	1	2	1
	2	3	2
	3	3	4
	4	1	4

UserGroup

	id	title	member_id	activity_id
▶	1	internship group	2	1
	2	party group	1	2
	3	party group	2	2

Sample Queries:

- *To select FOF of user 1.*

```
select friend_id
from relation
where user_id = 1 and status = 'friend'
union
select distinct friend_id
from relation
where friend_id <>1 and status = 'friend' and user_id in (select friend_id
                                    from relation
                                    where user_id = 1                                and status = 'friend') ;
```

	friend_id
▶	2
	3
	4

- *To view profile from DF of user 1.*

```
select id, username, real_name, age, city, last_login
from user
where id in (select friend_id
                    from relation
                    where status = 'friend' and user_id = 1)
order by last_login DESC;
```

	id	username	real_name	age	city	last_login
▶	2	amy001	Amy Wong	20	Paris	2019-04-13 11:13:50
	3	Jackac	Jack Williams	21	Beijing	2019-04-13 11:13:50

- *To view text post from FOF of user 1.*

```
select post_user, title, text_content, post_time
from post
where post_user in (select friend_id
from relation
where user_id = 1 and status = 'friend'
union
select distinct friend_id
from relation
where friend_id <>1 and status = 'friend' and user_id in (select friend_id
                                    from relation
                                    where user_id = 1 and status = 'friend') )
order by post_time DESC;
```

	post_user	title	text_content	post_time
▶	3	Ice hockey playing	Play with us!	2019-04-13 21:02:01
	4	Does someone want to play ice hockey?	I have some friends.	2019-04-13 21:01:06
	2	April 14th Diary	I got some videos.	2019-04-13 18:06:48
	3	Some helpful things	1.Google, 2.Facebook	2019-04-13 18:06:48
	3	personal diary	User 3 diary for today	2019-04-13 17:27:36
	4	title	awesome day	2019-04-13 17:27:36
	2	April 11th Diary	I post some selfies.	2019-04-13 11:39:23

- *To view multimedia post from DF of user 1.*

```
select post_user, title, text_content, post_time, photo_content, video_content, audio_content, post_time, location_id
from post left outer join photo on photo.id = post.photo_id left outer join video on video.id = post.video_id left outer join audio on audio.id = post.audio_id
where post_user in (select friend_id
from relation
where user_id = 1 and status = 'friend')
order by post_time desc;
```

post_user	title	text_content	post_time	photo_content	video_content	audio_content	post_time	location_id
3	Ice hockey playing	Play with us!	2019-04-13 21:02:01	NULL	BLOB	NULL	2019-04-13 21:02:01	5
2	April 14th Diary	I got some videos.	2019-04-13 18:06:48	NULL	BLOB	NULL	2019-04-13 18:06:48	3
3	Some helpful things	1.Google, 2.Facebook	2019-04-13 18:06:48	NULL	NULL	NULL	2019-04-13 18:06:48	2
3	personal diary	User 3 diary for today	2019-04-13 17:27:36	NULL	NULL	NULL	2019-04-13 17:27:36	1
2	April 11th Diary	I post some selfies.	2019-04-13 11:39:23	BLOB	NULL	NULL	2019-04-13 11:39:23	1

- *To view multimedia post from FOF of user 1.*

```
select post_user, title, text_content, post_time, photo_content, video_content, audio_content, post_time, location_id
from post left outer join photo on photo.id = post.photo_id left outer join video on video.id = post.video_id left outer join audio on audio.id = post.audio_id
where post_user in (select friend_id
from relation
where user_id = 1 and status = 'friend'
union
select distinct friend_id
from relation
where friend_id <>1 and status = 'friend' and user_id in (select friend_id
                                from relation
                                where user_id = 1 and status = 'friend') )
order by post_time desc;
```

post_user	title	text_content	post_time	photo_content	video_content	audio_content	post_time	location_id
3	Ice hockey playing	Play with us!	2019-04-13 21:02:01	NULL	BLOB	NULL	2019-04-13 21:02:01	5
4	Does someone want to play ice hockey?	I have some friends.	2019-04-13 21:01:06	NULL	NULL	NULL	2019-04-13 21:01:06	5
2	April 14th Diary	I got some videos.	2019-04-13 18:06:48	NULL	BLOB	NULL	2019-04-13 18:06:48	3
3	Some helpful things	1.Google, 2.Facebook	2019-04-13 18:06:48	NULL	NULL	NULL	2019-04-13 18:06:48	2
4	title	awesome day	2019-04-13 17:27:36	NULL	NULL	BLOB	2019-04-13 17:27:36	2
3	personal diary	User 3 diary for today	2019-04-13 17:27:36	NULL	NULL	NULL	2019-04-13 17:27:36	1
2	April 11th Diary	I post some selfies.	2019-04-13 11:39:23	BLOB	NULL	NULL	2019-04-13 11:39:23	1

- *To view multimedia post from FOF of user 2 which contains words “ice hockey”.*

```
select post_user, title, text_content, post_time, photo_content, video_content, audio_content, post_time, location_id
from post left outer join photo on photo.id = post.photo_id left outer join video on video.id = post.video_id left outer join audio on audio.id = post.audio_id
where text_content like '%ice hockey%' or title like '%ice hockey%'and post_user in (select friend_id
from relation
where user_id = 2 and status = 'friend'
union
select distinct friend_id
from relation
where friend_id <>2 and status = 'friend' and user_id in (select friend_id
                                from relation
                                where user_id = 2 and status = 'friend') )
order by post_time desc;
```

post_user	title	text_content	post_time	photo_content	video_content	audio_content	post_time	location_id
3	Ice hockey playing	Play with us!	2019-04-13 21:02:01	NULL	BLOB	NULL	2019-04-13 21:02:01	5
4	Does someone want to play ice hockey?	I have some friends.	2019-04-13 21:01:06	NULL	NULL	NULL	2019-04-13 21:01:06	5

- *To view multimedia post from DF of user 1 from last week.*

```
select post_user, title, text_content, post_time, photo_content, video_content, audio_content, post_time, location_id
from post left outer join photo on photo.id = post.photo_id left outer join video on video.id = post.video_id left outer join audio on audio.id = post.audio_id
where post_time>unix_timestamp(NOW())-3600*24*7 and post_user in (select friend_id
from relation
where user_id = 1 and status = 'friend')
order by post_time desc;
```

post_user	title	text_content	post_time	photo_content	video_content	audio_content	post_time	location_id
3	Ice hockey playing	Play with us!	2019-04-13 21:02:01	NULL	BLOB	NULL	2019-04-13 21:02:01	5
2	April 14th Diary	I got some videos.	2019-04-13 18:06:48	NULL	BLOB	NULL	2019-04-13 18:06:48	3
3	Some helpful things	1.Google, 2.Facebook	2019-04-13 18:06:48	NULL	NULL	NULL	2019-04-13 18:06:48	2
3	personal diary	User 3 diary for today	2019-04-13 17:27:36	NULL	NULL	NULL	2019-04-13 17:27:36	1
2	April 11th Diary	I post some selfies.	2019-04-13 11:39:23	BLOB	NULL	NULL	2019-04-13 11:39:23	1

- *To view all liked location from DF of user 1.*

```
select location.id, location.GPS_coordinate, location.building, location.room
from location join likelocation on location.id = likelocation.location_id
where user_id in (select friend_id
from relation
where user_id = 1 and status = 'friend');
```

id	GPS_coordinate	building	room
1	41°21'N 2°12'E	Watson building	302
2	39°24'N 1°42'E	Metrotech building	501
4	47°21'N 4°12'E	Game lab	208

✦ Define Stored Procedures

- *Login Check*

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `login_check`(IN username_user varchar(64),
IN password_user varchar(64))
BEGIN
        IF EXISTS(SELECT * FROM USER WHERE username = username_user AND password =
password_user) then
                SELECT username,real_name, age, city, last_login FROM user
                WHERE username = username_user AND password = password_user;
                update user set last_login = current_timestamp()
                 where username = username_user;
   ELSEIF EXISTS(SELECT * FROM USER WHERE username = username_user AND password
<> password_user) then
                SET @message_text = CONCAT('Login failed for \'', username_user, '\': incorrect
password');
                SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = @message_text;
        ELSE
                SET @message_text = CONCAT('Login failed for \'', username_user, '\': incorrect
username');
                SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = @message_text;
   END IF;END
```

```
1 • call SocialNetwork.login_check('tom0125', '12345');
```

username	real_name	age	city	last_login
tom0125	Tom Johnson	24	New York	2019-04-13 11:13:50

```
Error Code: 1644. Login failed for 'tom': incorrect username
Error Code: 1644. Login failed for 'tom0125': incorrect password
```

- *Register user*

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `user_create`(IN name CHAR(64), IN pass
CHAR(64))
BEGIN
        IF (SELECT COUNT(user.id) FROM user WHERE username = name) > 0 THEN
                SET @message_text = CONCAT('User \'', name, '\' already exists');
                SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = @message_text;
        ELSE
                INSERT INTO user(username, password) VALUES (name, pass);
                CALL user_auth(name, pass);
        END IF;
End
```

```
Error Code: 1644. User 'tom0125' already exists
```

- *Check comment legitimacy*

To select friend or friend of friend based on visibility and return the legitimacy.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `comment_legitimacy`(IN from_id varchar(64),
IN to_id varchar(64),IN post_id int, In content varchar(500),In visibility int)
BEGIN
        IF EXISTS(SELECT * FROM relation WHERE user_id = from_id AND friend_id = to_id AND
status = 'friend' union SELECT * FROM relation WHERE user_id = to_id AND friend_id in (select
friend_id from relation where user_id = to_id and status = 'friend')  AND visibility > 1) then
                insert into comment(from_id, to_id, post_id, content)
    value
    (from_id, to_id, post_id, content);
    set @username = (select username from user where id = from_id);
    SET @message_text = CONCAT('Comment from user\'', @username, '\' is created');
                select @message_text;
  ELSE
                set @username = (select username from user where id = from_id);
                SET @message_text = CONCAT('Comment failed for user\'', @username, '\': You
have no authentication to comment!');
                SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = @message_text;
        END IF;
END
```

Correct situation:

```
call SocialNetwork.comment_legitimacy('1', '4', 7, 'Hello', 2);
```

@message_text
Comment from user'tom0125' is created

Thus new row in the comment table:

	3	1	4	7	Hello	2019-04-14 13:36:26

Wrong situation:
Because user 1 and user 4 are FOF, but post.visibility of user 4's post is 1, which is limited to DF. So the comment is not permitted. Then Error code occurs.

```
call SocialNetwork.comment_legitimacy('1', '4', 7, 'wrong input', 1);
```

Error Code: 1644. Comment failed for user'tom0125': Your have no authentication to comment!

- *Establish friendship*

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `check_relationship`(IN user_id_new int, IN friend_id_new int, In visibility int)
BEGIN
        set @username1 = (select username from user where id = user_id_new);
        set @username2 = (select username from user where id = friend_id_new);
   IF user_id_new = friend_id_new then
                SET @message_text = 'Cannot send friendship request to oneself!';
                SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = @message_text;
        ELSEIF EXISTS(select * from relation where user_id = friend_id_new and friend_id = user_id_new and status = 'pending') then
                insert into relation(user_id, friend_id, status, visibility)
      value
      (user_id_new, friend_id_new, 'friend', visibility);
      update relation set status = 'friend' where user_id = friend_id_new and friend_id = user_id_new;

      SET @message_text = CONCAT('Friendship between user\'', @username1, '\' and \'', @username2, '\' is created');
                select @message_text;

   ELSEIF EXISTS(select * from relation where user_id = friend_id_new and friend_id = user_id_new and status = 'friend') then
      SET @message_text = CONCAT('Friendship between user\'', @username1, '\' and \'', @username2, '\' is already established');
                SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = @message_text;

        ELSEIF EXISTS(select * from relation where user_id = user_id_new and friend_id = friend_id_new and status =‘pending') then
      SET @message_text = CONCAT('Already sent friendship request from\'', @username1, '\' and \'', @username2, ‘\'.');
                SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = @message_text;

        ELSE
                insert into relation(user_id, friend_id, status, visibility)
      value
      (user_id_new, friend_id_new, 'pending', visibility);
      SET @message_text = CONCAT('Friendship request from user\'', @username1, '\' to \'', @username2, '\' is sent');
                select @message_text;
        END IF;
END
```

There are totally five situations:

1. user A is requesting friendship user B:

1	5	pending	2

```
call SocialNetwork.check_relationship(1, 5, 2);
```

@message_text
Friendship request from user'tom0125' to 'jerry23' is sent

2. user A is already requested friendship towards user B and user B is accepting it.

1	4	pending	3

```
call SocialNetwork.check_relationship(4, 1, 3);
```

Then result will be

@message_text
Friendship between user'Ding' and 'tom0125' is created

4	1	friend	3
1	4	friend	3

3. user A is sending request to himself.

```
call SocialNetwork.check_relationship(5, 5, 3);
```

↓

Error Code: 1644. Cannot send friendship request to oneself!

4. user A sent friendship request before but A send another request

```
call SocialNetwork.check_relationship(1, 5, 3);
```

↓

Error Code: 1644. Already sent friendship request from'tom0125' and 'jerry23'

5. user A and user B are already friends but A send another request

```
call SocialNetwork.check_relationship(1, 2, 3);
```

↓

Error Code: 1644. Friendship between user'tom0125' and 'amy001' is already established

- *Search Content*

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `search`(IN keyword varchar(100),IN user_id_in int)
BEGIN
	IF EXISTS(
		select *
		from post
		where title like concat('%',keyword,'%') or text_content like concat('%',keyword,'%')
	and post_user in (select friend_id from relation where user_id = user_id_in and status = 'friend')
		union
		select *
		from post
		where title like concat('%',keyword,'%') or text_content like concat('%',keyword,'%') and visibility > 2 and
post_user in
	(select distinct friend_id from relation where status = 'friend' and post_user in (select friend_id from relation where user_id =
user_id_in and status = 'friend') )
	)
	then
			select post_user, title, text_content, post_time, photo_content, video_content, audio_content, location_id
			from post left outer join photo on photo.id = post.photo_id left outer join video on video.id = post.video_id
left outer join audio on audio.id = post.audio_id
			where title like concat('%',keyword,'%') or text_content like concat('%',keyword,'%')
		and post_user in (select friend_id from relation where user_id = user_id_in and status = 'friend')
			union
		select post_user, title, text_content, post_time, photo_content, video_content, audio_content, location_id
			from post left outer join photo on photo.id = post.photo_id left outer join video on video.id = post.video_id
left outer join audio on audio.id = post.audio_id
			where title like concat('%',keyword,'%') or text_content like concat('%',keyword,'%') and visibility > 2
		and post_user in (select distinct friend_id from relation where status = 'friend' and user_id in (select friend_id from relation
where user_id = user_id_in and status = 'friend'))
		order by post_time desc;
   ELSE
		SET @message_text = CONCAT('There is no post contains \'', keyword, '\'.');
		SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = @message_text;
	END IF;
END
```

Result of post search:
If we search ice hockey in post and select search user id is 2:
Because user 2 is friend of user 1 and user 4, and FOF of user 2 is 3. So user 2 can search all the post contains 'ice hockey', the result will be:

post_user	title	text_content	post_time	photo_content	video_content	audio_content	location_id
3	Ice hockey playing	Play with us!	2019-04-13 21:02:01	NULL	BLOB	NULL	5
4	Does someone want to play ice hockey?	I have some friends.	2019-04-13 21:01:06	NULL	NULL	NULL	5

But if we use user 1 to search, the result post list will be:

```
call SocialNetwork.search('ice hockey', 1);
```

100% 42:1

Result Grid Filter Rows: Search Export:

post_user	title	text_content	post_time	photo_content	video_content	audio_content	location_id
3	plasdf with use	ice hockey now!	2019-04-15 20:27:47	BLOB	NULL	NULL	5
3	Ice hockey playing	Play with us!	2019-04-13 21:02:01	NULL	BLOB	NULL	5
4	Does someone want to play ice hockey?	I have some friends.	2019-04-13 21:01:06	NULL	NULL	NULL	5

It is because post with text_content in ice hockey's visibility is 1, so it will only be seen by direct friend and user 2 is not direct friend of post 3. So this query is correct.
If we search for a text which is not contained by all its visible post, it will return an error

```
call SocialNetwork.search('Facebook', 2);
```

Error Code: 1644. There is no post contains 'Facebook'.

user 1 will see this content because user 5 has a friendship with user 1:

```
call SocialNetwork.search('Facebook', 1);
```

100% 40:1

Result Grid Filter Rows: Search Export:

post_user	title	text_content	post_time	photo_content	video_content	audio_content	location_id
5	Some helpful things	1.Google, 2.Facebook	2019-04-13 18:06:48	NULL	NULL	NULL	2

✦Summary of database design

In this schema, I designed a university students social network database.
For the user login part, I stored the username and password in the user table, if a user login with the wrong password, then it will not return information of users. If login successes, then the user's profile will be returned. It will contain real_name, age, city and last login information. For last login information, I set up a update timestamp to update the time when user login.
Then a user can public a post, organize or attend an event, write comment to others, mark like to locations, define relationship with other users.

- For publish a post, user can define title, text_content, location_id, and decide whether or not to attach a photo, a video, a audio. It will attach three tables in order to get information about the attached file. The visibility is defined by int number, 1 for

direct friend, 2 for FOF (friend of friend), 3 for all. The post_user and post_time will be generate automatically.

- For publish an event, user can publish an activity into activity table, specify the event description, event_time and location_id for an event. To join an event, a user must join in a usergroup for this activity, then the usergroup and activity table will join to query for the specific activity for a user to join in. This group will also represent for group of different organizations and courses.
- For write comment action, user can only browse the specific content from his friends, his FOF or all users, then it will use a comment table to record the comment content, the id this comment from and id for post user and the id for post content, it will always use a constraint to check if the post_user and from_id are the relationship the visibility specified. The comment_time will generate automatically.
- For like a location, the like location table will record the location some users like, then if someone wants to find out his DF or FOF likes, then it will use a query to relation and like location to find out the location liked by friends. It will also effect the location like counter, which is basically count the location_id from like location table and display it.
- For relationship with others, I use a relation table to store the request line of relationship. If one user A request of friendship with user B, then relation table will create one row to indicate the request, and status will be pending. User A can also specific the visibility of this relationship. If user B get the relationship request, and reply as accept, then the status of both rows will become friend. It will create totally two lines to indicate the friendship: A→B, B→A. The visibility of this friendship will keep as each one of them's visibility. It will also generate the time of request and response.

Other parts like timeline, post visibility to users will perform with queries, like the sample queries above. It will check the visibility of post with friendship to make sure the comment will never have wrong input.

Part 2 Website Design

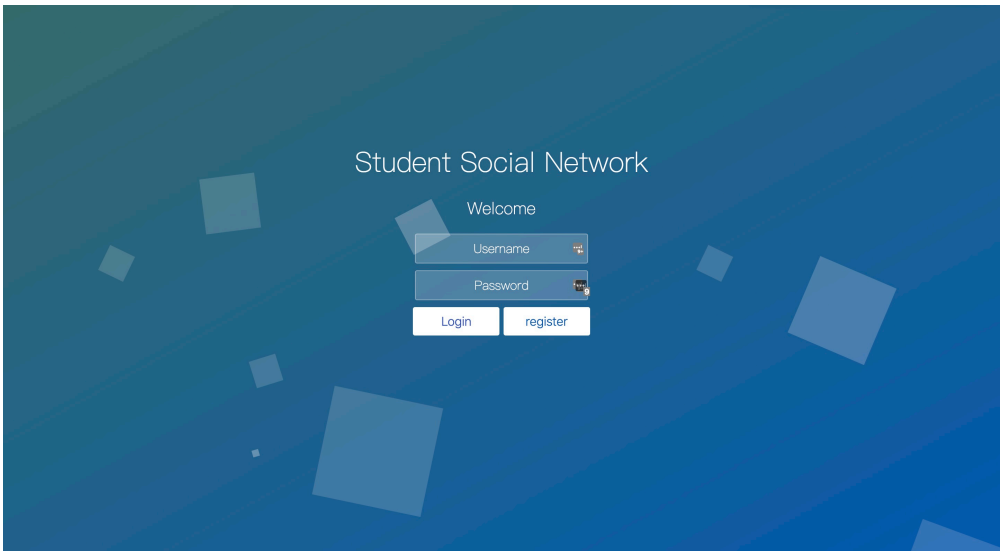
In this part of document, I will explain my website design and overall website design tools.

I have totally 7 pages in my website, they are login.html (with login.php as back-end support), register.html (with register.php as back-end design), index.php (with submit.php, query.php, likepost.php as back-end), post.php, friend.php (with friend request.php as back-end), profile.php, search.php (header.html and footer.html are as prefix and suffix for all pages).

I will explain them with screenshot for every page.

1. Login.html

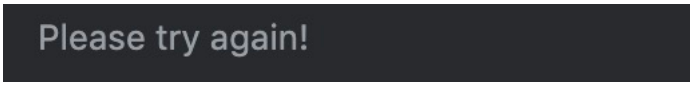
This page is mainly for login. When users visit this page, page will be as below.



For input text areas, users are supposed to input the username and password if they are already register, otherwise, they should click the register button directly to visit the register page. If the username and password are correct, then it will login and redirect to main page.

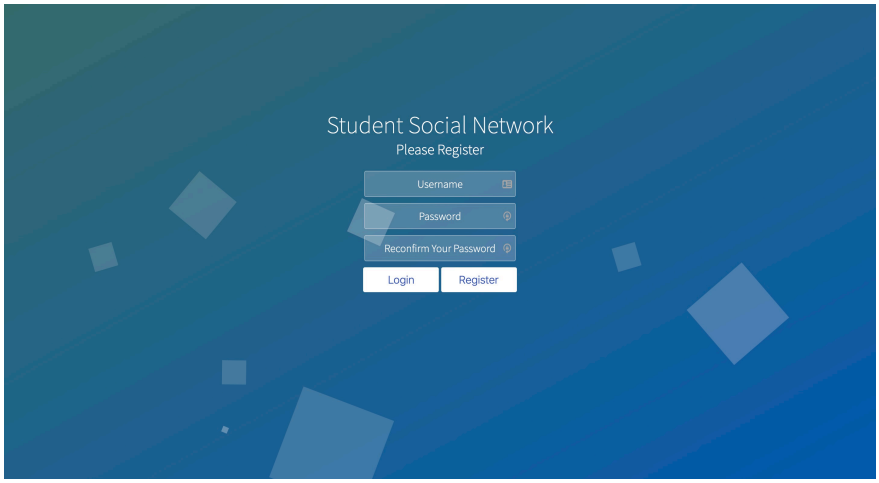
If login success, this message will pop up.

Otherwise, the error message will pop up.



2. Register.html

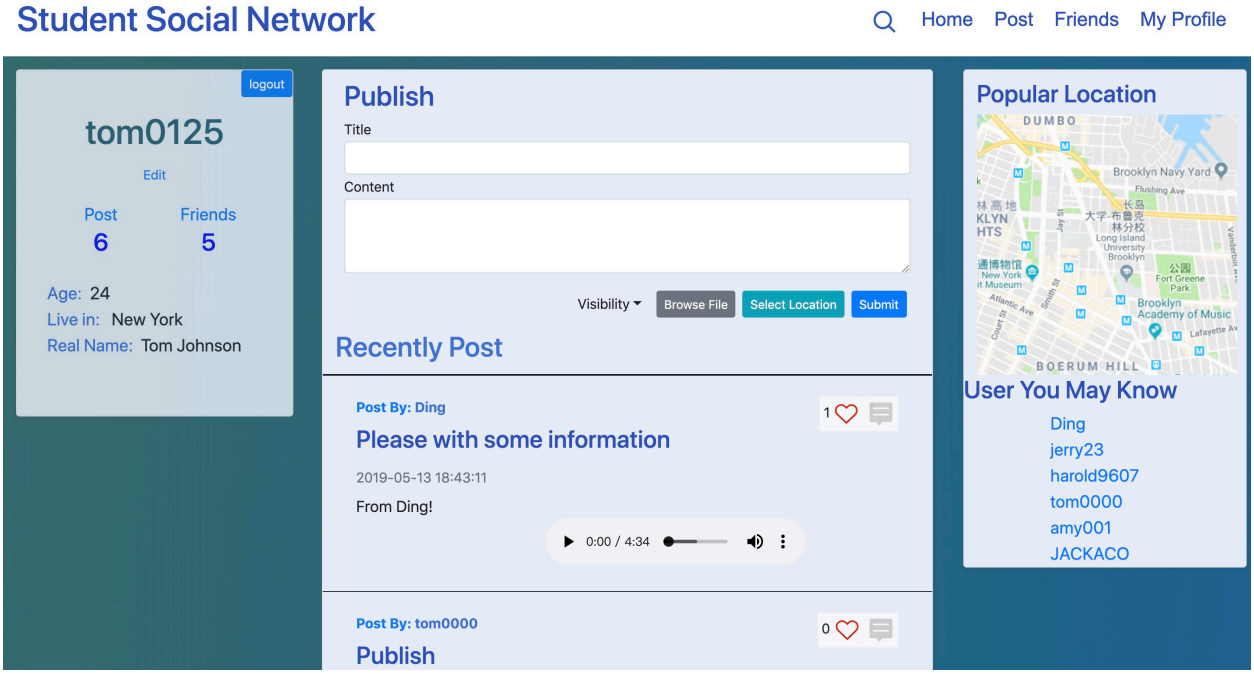
This page is mainly for register a new user. When user do not have an account, this page will call database stored procedure to create a new user. This page is like below.



Users are supposed to create a unique username and give password for two times to make sure the password is same. Then the alert message will pop up to show if the register is success or not.

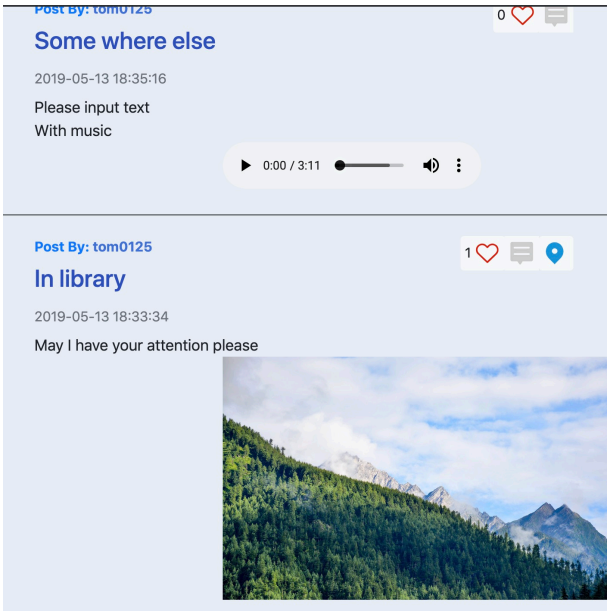
3. Main page (index.php)

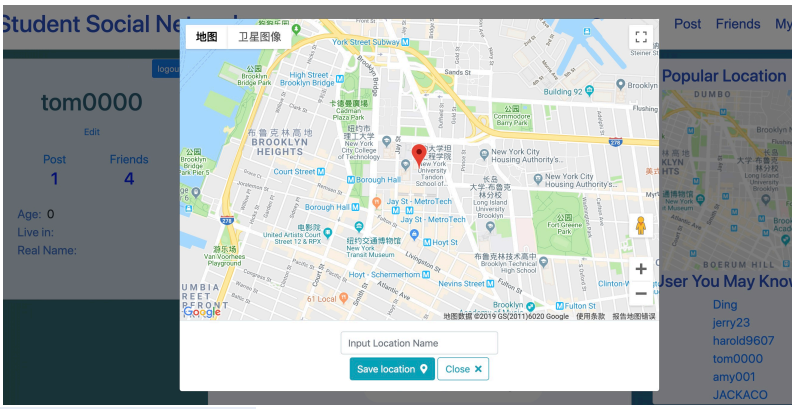
I am using the php language for all the html pages. In this page, users can see many informations.

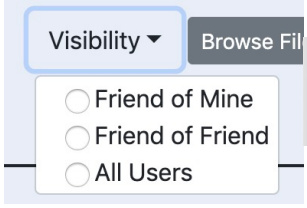


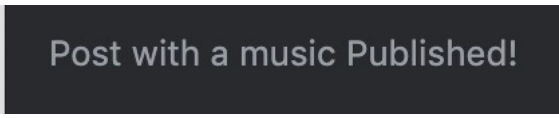
There are four parts to show different information.

- The upper part is the navigation bar. From where user can click on the ‘search’ button, the ‘home’ button, the ‘post’ button, the ‘friend’ button, the ‘my profile’ button to get to different pages.
- The left part is to show the user’s information, the logout button is used to log out from the website, then it shows username, post number, friend number, user age, user city, user real name for the login user. Users can click on the edit button to the profile page to change some information about themselves. The ‘Post’ button and ‘Friends’ button are also clickable, ‘Post’ button will lead to the post page where shows what did they post, ‘Friends’ button will lead to the Friend page shows the friend request and current friends.
- The middle part of this webpage is used for publish and view the post from friends and FOF (friends of friends).Some screenshots are listed below.

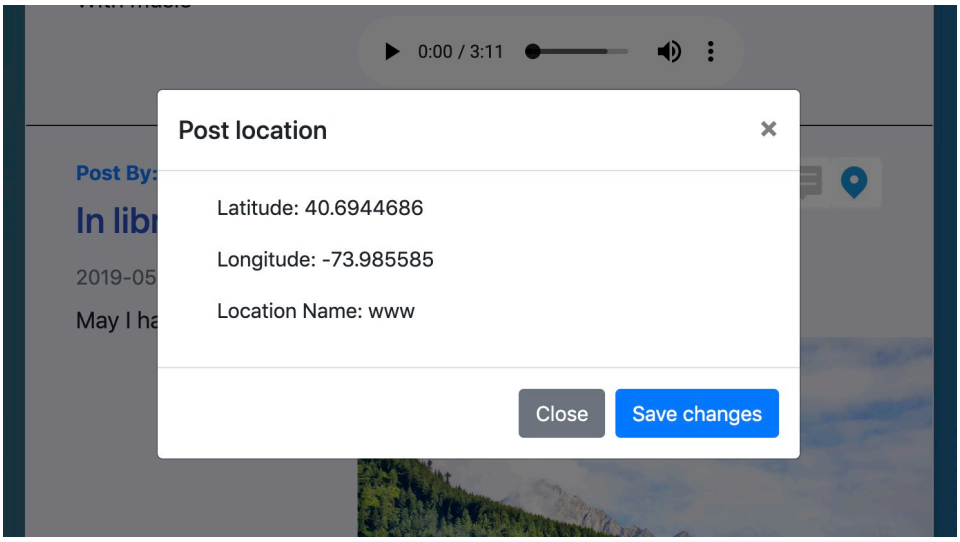








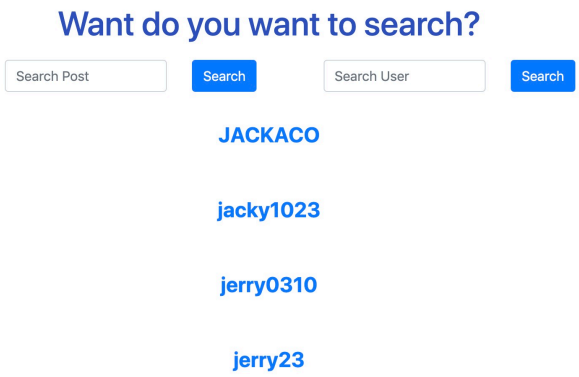
- For publish part, we can use the select location button to get a new pop-up window, which can select current location's GPS information, with the user input location name, we can store it in the position database schema. Then for the input text area, we can input the title and content for the post. For the visibility button, we can choose users group to see. For Browse file part, we can upload file to the post. When click on the submit button, we can see the new post is as below (contain music).
- The Recently post page will show the FOF and Friends' post information, including who post it, title, time, text content for the post, multimedia information will show in the bottom (music, picture, video). In the upright will include like, comment, location information. If user uploaded post with location, it will show the geolocation coordinates and name of the post location.



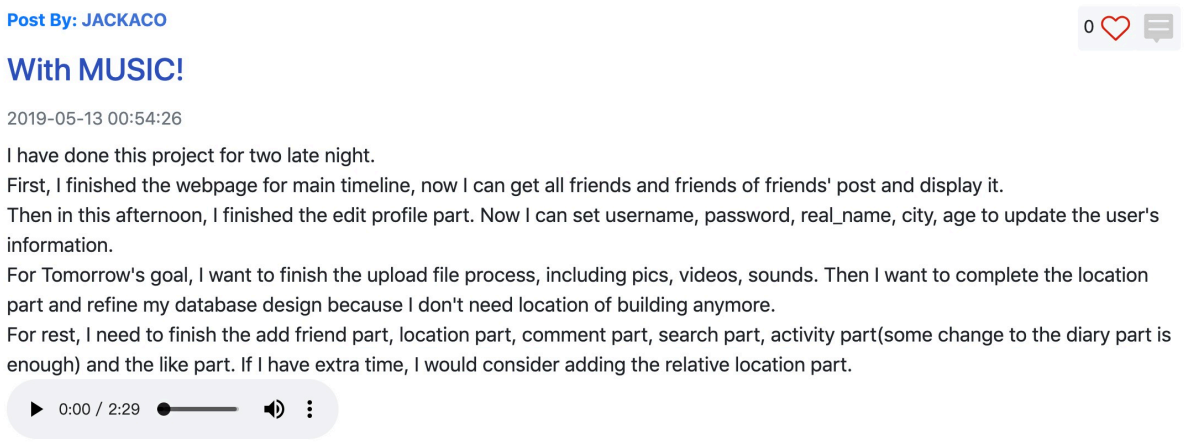
- When click on the like button, the like number will add one. Comment part is not complete, but I add comment schema and comment back-end part for this project.

• On the right side bar, the popular location map will show the most popular location in the map and show how many posts are published in this map. Then in the bottom of the bar, it will show some FOF friend of mine, but not including my friends. We also can click on the link to the friends' page to follow and make friend with them.

4. Search page (search.php)

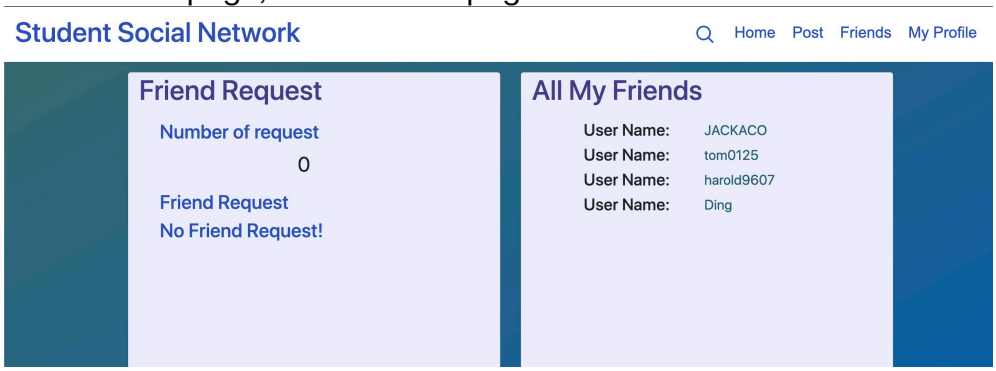


- When input some strings to search user area, the page will pop out the username with this string, like I input ‘j’ in this page, and all the username with ‘j’ are showed, and we can use this link to visit the page of this user.
- Search post part is at left side, when input string like ‘k’, with a list of all the post with string ‘k’ will pop up, including all the information like the main page.



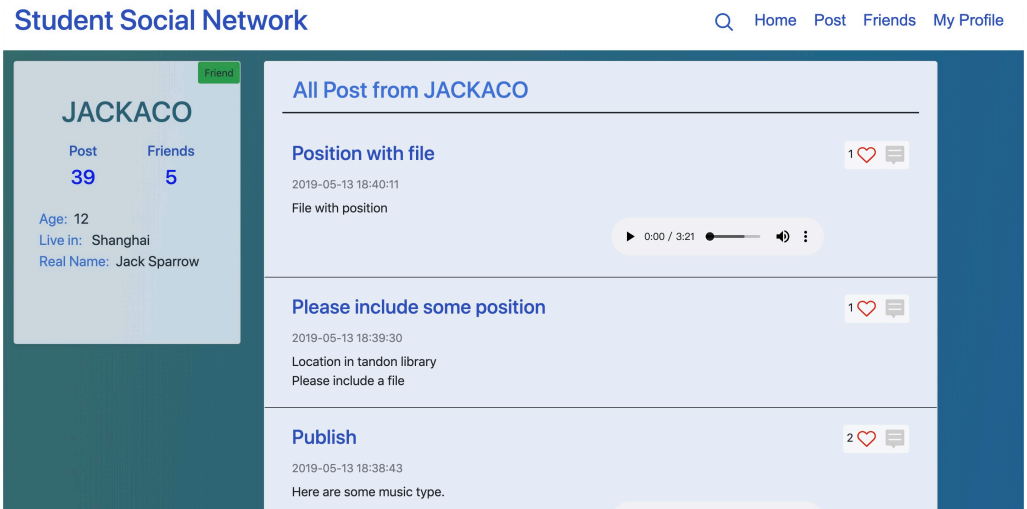
5. Friend Page (friend.php)

Suppose we are user A, when we click the friend page in the navigation bar or the left side bar in the main page, we will see a page like this:



It lists the friends I have related with, and the friend request from other users, the friend link are also activated.

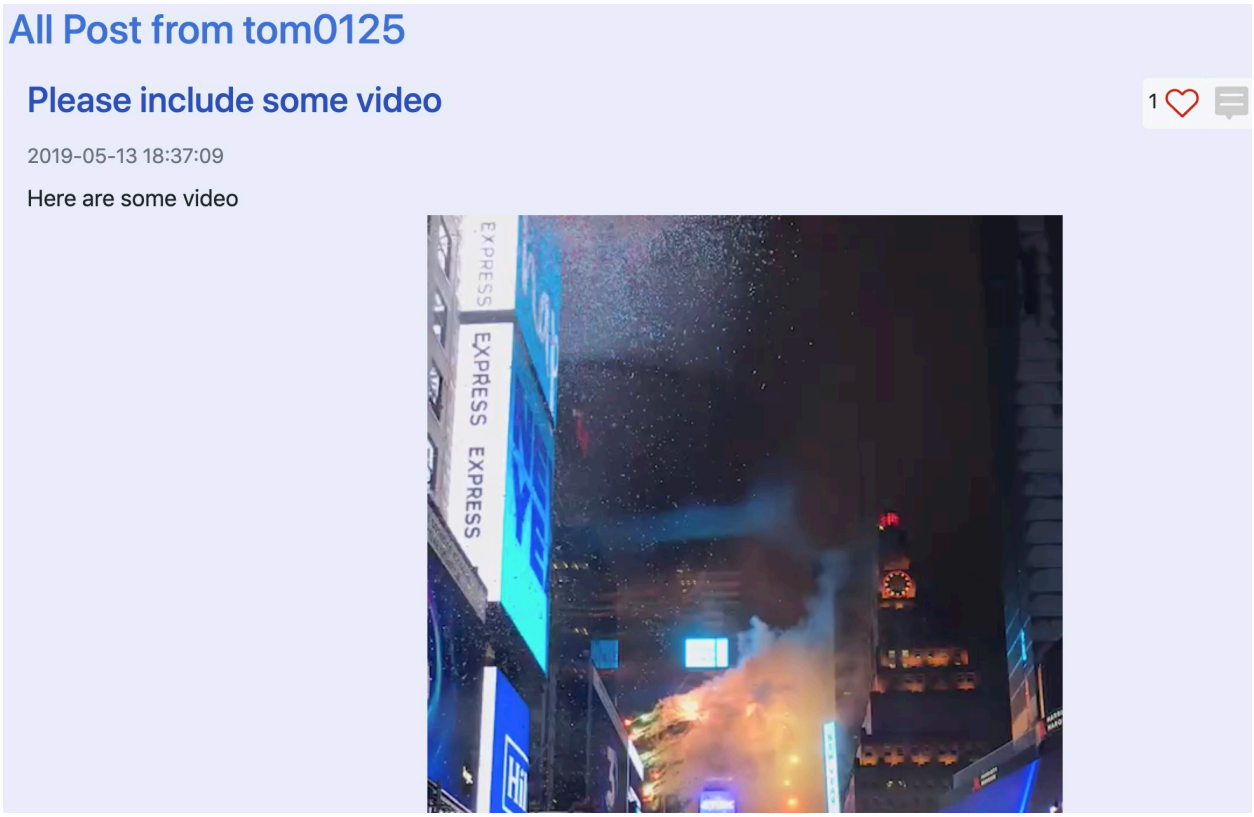
If we click on any user’s link from main page or the friend page, we can jump to his personal info and post page. Like below. If two users are friends, the left button is friend instead of (follow (didn’t send request) or followed (sent request but not agree)).



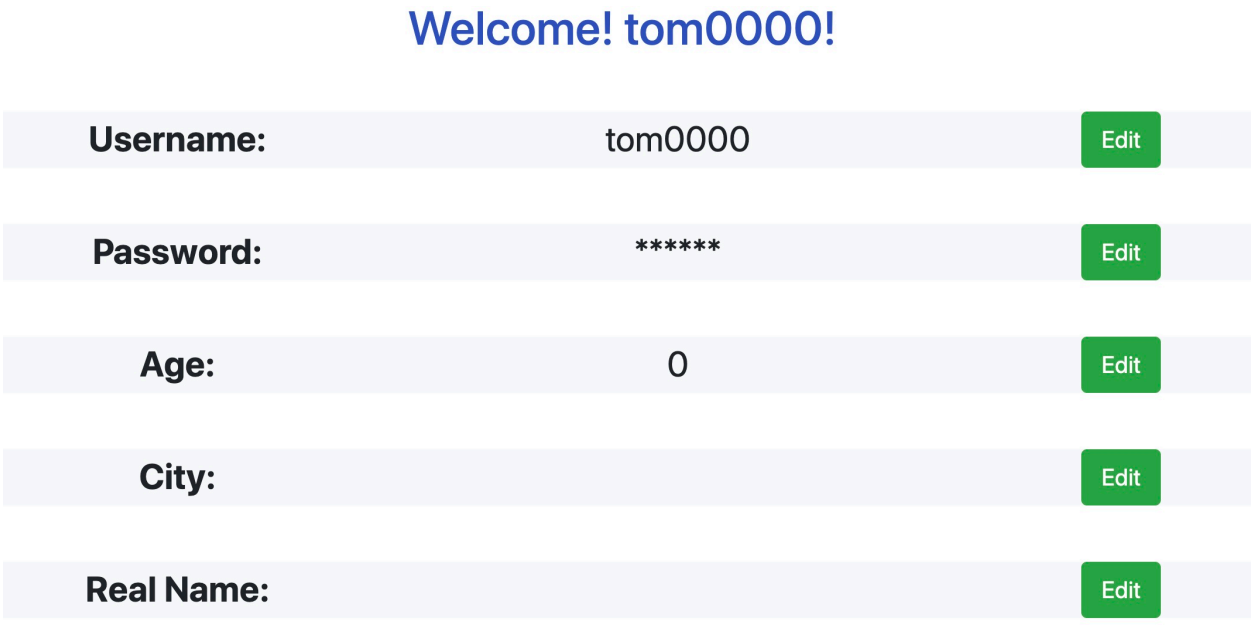
It will show information of the selected user, and post from him. It is different from main page is that the link in the post and friends of the left side bar is not clickable, so we can't see his friends.

6. Post page

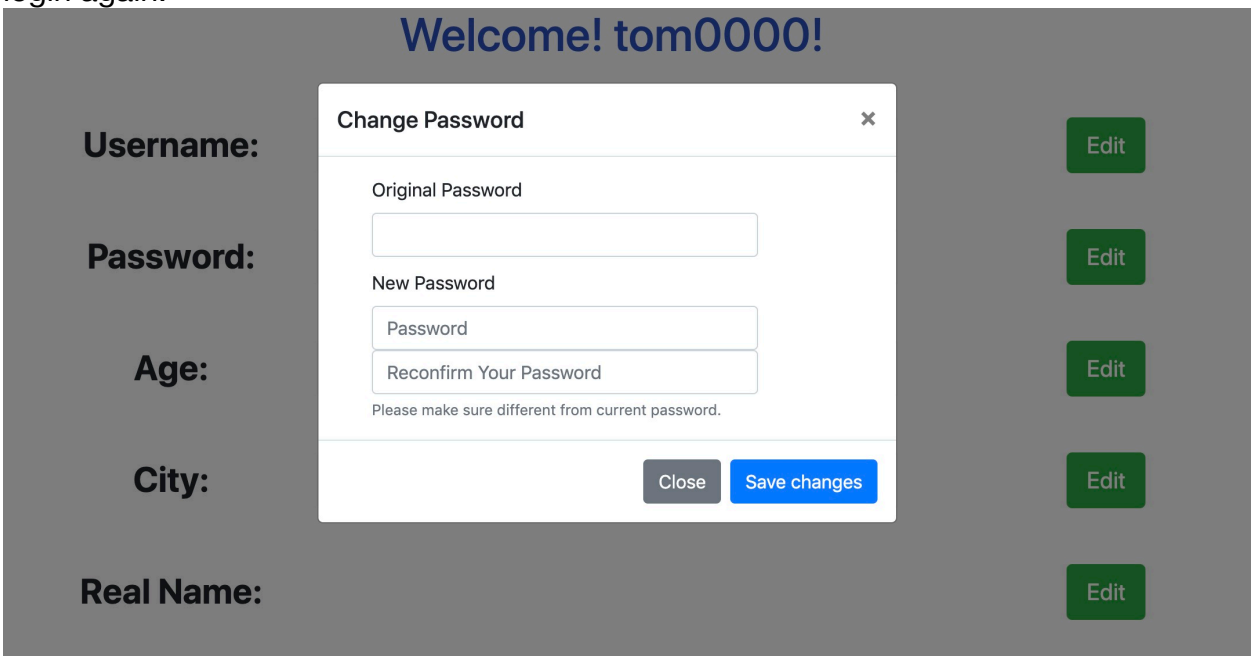
This page show the user his own page of post, it will show all the post he published and give user his post with comment and like condition.



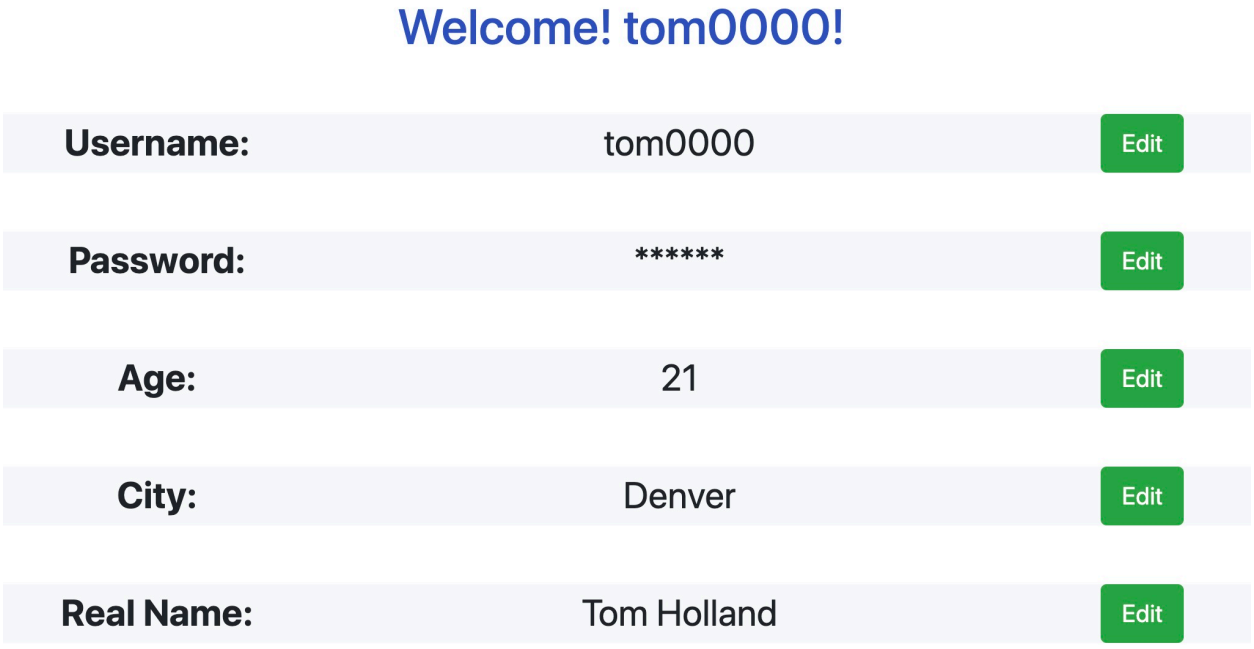
7. Profile page



When we register for the first time, the profile page will be like this, and we can click on the edit button to change. The change to username part will lead to logout and need to login again.



For all change pages, it will pop up a window to edit the information, the password part need to reconfirm the password. After the editing, the page will be like:



The demonstration for the project is finished. Thank you for your patience to read. Wish you a happy day!