

```
DROP DATABASE photoshare;
CREATE DATABASE IF NOT EXISTS photoshare;
USE photoshare;
DROP TABLE IF EXISTS Pictures CASCADE;
DROP TABLE IF EXISTS Users CASCADE;
```

```
CREATE TABLE Users (
    user_id int4  AUTO_INCREMENT,
    email varchar(255) UNIQUE,
    password varchar(255),
    fName varchar(255),
    lName varchar(255),
    DOB DATE,
    hometown varchar(255),
    gender varchar(255),
    CONSTRAINT users_pk PRIMARY KEY (user_id)
);
```

```
CREATE TABLE Albums(
    albumID      INTEGER AUTO_INCREMENT,
    albumName    CHAR(20) NOT NULL,
    albumDate    DATE,
    albumOwnedBy int4,
    PRIMARY KEY (albumID),
    FOREIGN KEY (albumOwnedBy)
REFERENCES Users (user_id)
);
```

```
CREATE TABLE Pictures
(
    picture_id int4  AUTO_INCREMENT,
    user_id int4,
    imgdata longblob ,
    caption VARCHAR(255),
    belongs INTEGER NOT NULL,
    INDEX upid_idx (user_id),
    CONSTRAINT pictures_pk PRIMARY KEY (picture_id)
);
```

```
CREATE TABLE FriendsWith (
    userID1      INTEGER,
```

```
        userID2      INTEGER,  
        PRIMARY KEY (userID1, userID2),  
        FOREIGN KEY (userID1) REFERENCES Users (user_id),  
        FOREIGN KEY (userID2) REFERENCES Users (user_id),  
        CONSTRAINT selfFriend CHECK (userID1 != userID2)  
);
```

```
CREATE TABLE Likes(  
    userID      INTEGER,  
    photoID     INTEGER,  
    PRIMARY KEY (userID, photoID),  
    FOREIGN KEY (userID)  
REFERENCES Users (user_id),  
    FOREIGN KEY (photoID)  
REFERENCES Pictures (picture_id)  
);
```

```
CREATE TABLE Tags(  
    tagDescription CHAR(20),  
    PRIMARY KEY (tagDescription)  
);
```

```
CREATE TABLE taggedWith(  
    photoID INTEGER,  
    tagDescription CHAR(20),  
    PRIMARY KEY (photoID, tagDescription),  
    FOREIGN KEY (photoID)  
        REFERENCES Pictures(picture_id),  
    FOREIGN KEY (tagDescription)  
        REFERENCES Tags(tagDescription)  
);
```

```
CREATE TABLE Comments(  
    commentID      INTEGER AUTO_INCREMENT,  
    commentText     CHAR(100) NOT NULL,  
    commentDate     DATE,  
    commentOwnedBy  INTEGER,  
    commentedUnder  INTEGER NOT NULL,  
    PRIMARY KEY (commentID),  
    FOREIGN KEY (commentOwnedBy)  
REFERENCES Users (user_id),
```

User Management

Becoming a registered user.

- Before uploading the photos, the user first needs to get registered. Attributes including email and password are required, and attributes including first name, last name, date of birth, hometown, and gender are optional.
- An email address can be only registered once.

Adding and Listing Friends.

- This feature can be only accessed by the registered user. When an unregistered user clicked on "Find and View Friends!", he/she will be directed to "Create an account" page.
- The search is done on the email. If the user enters text other than email address into the search box, clicking on "submit" will redirect the user back to the "Friend Search" page. If no results are returned from the database (which means the email hasn't been registered), clicking on "submit" will also redirect the user back to the "Friend Search" page.
- Users can search for friends in the system. If a friend has been found, the system will print the results. The user can add the friend by clicking on the hyperlink beside. After adding the friend, it will appear in the Current Friend List.

Search Users:

Enter User's Email:

Select a friend:

- Meichen Dong jazzmyn@bu.edu [ADD FRIEND](#)

Current Friends

Meichen Dong jazzmyn@bu.edu

- If the user tried to add the same friend, clicking on "submit" will direct the user to the "Current Friends".

User activity

- We measure the contribution of a user as the number of photos they have uploaded plus the number of comments they have left for photos belonging to other users. The top 10 users should be reported.

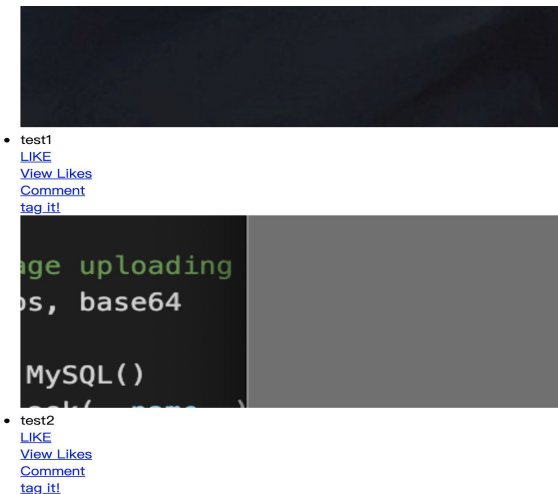
Album and Photo Management

Photo and Album Browsing

- Photos and albums are public. Both the registered users and the unregistered users are able to browse them by clicking on the “View All Albums” and “View All Photos”. This implementation was done using a search query, which returns everything in the photo and album database. Only users are allowed to access “View Your Albums” and “View Your Photos”. If the unregistered user clicks on these two hyperlinks, he will be redirected to the “create account” page.

Photo and album creating

- Registered users are able to create albums and upload photos. Before uploading a photo, the user must create an album first. This is a limitation of our system. When uploading a photo, the user must fill in the “caption” box and the “album” box, or the photo can’t be uploaded.
- Users can delete both the albums and the photos they owned. If nonempty album is deleted, the photos inside it will be deleted. Here are the page for photo list and album list:



- 1
[VIEW DELETE ALBUM](#)
- 2
[VIEW DELETE ALBUM](#)
- [Your Album List](#)
- [Home](#)

Comments

Leaving comments

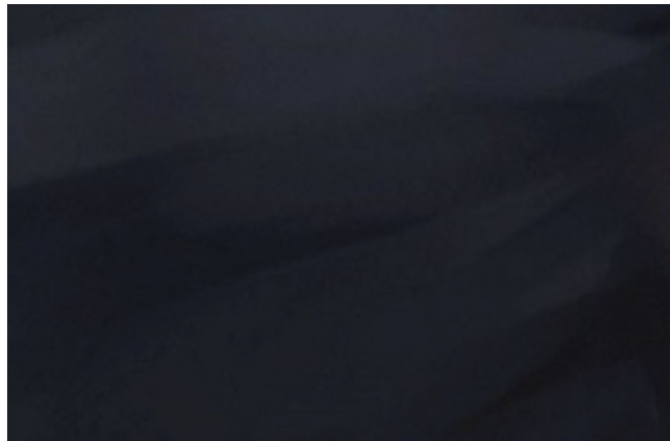
- Users are allowed to leave comments. But users cannot leave comments for their own photo.
- The limitation is that unregistered users cannot leave comments.

Like functionality

- Users can like the photos by clicking on the “LIKE” hyperlink. Then the user will be redirected to the “Like” page, where they can view how many likes they have. When clicking on “View Likes”, the system will print the name and email of the users who like the photo.

Selected Photo:

• 1



Total Likes: 2

- Meichen Dong 2571102516@qq.com
- test1@gmail.com
- Go back to [all photos?](#)

- The limitation is that only registered users can like the photo. If an unregistered user likes a photo, errors will be thrown. For future improvements, we should enable unregistered users like the photo.

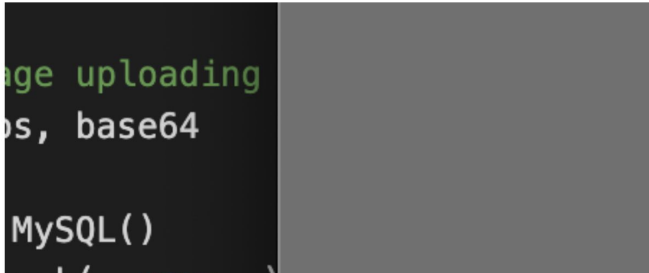
Search on comment

- The user can specify a text query (one or more words) and the system should find the users that have created comments that exactly match the query text.
- The system will return these pictures ordered by the number of comments that match the query for each such user in descending order.

- Limitations: there are several blank lines below the “Comments”, for future improvements, we will make the page format better

Selected Photo:

- test2



Find photos with a comment

enter comment:

Post a Comment::

Comments:

- comment by user test 3
- comment by user test 2
- great (comment by user test1)
- good
- good,good
- bad bad

Tag management

View your photos by tag name

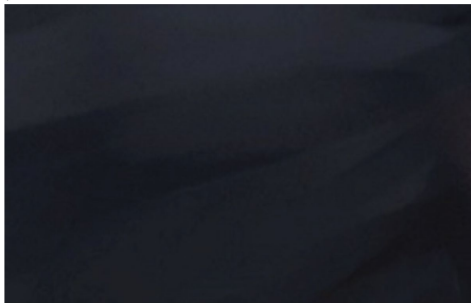
- Tag can be added to a photo by the user, each photo can have multiple tags. Tags are presented as hyperlink. When tags are clicked, the photo with the tag will be listed.

Search Photos By Tag:

Enter tag to search by:

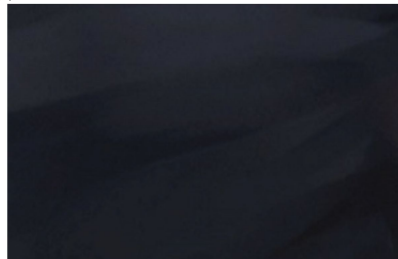
Here are your photos

- 1



[LIKE](#)
[View Likes](#)
[Comment](#)

- 1



Post a tag::

tags:

- black
- boston
- newyork

View all photos by tag name:

- When clicking on the “View all tags”, the user is able to view all the photos associated with this tag. This feature is available for all the users.

Viewing the most popular tags:

- When clicking on the popular tags hyperlink, the top 5 most popular tags will be returned.

Tag Search:

- Users can access this feature by clicking on “Search photo by tag”. Both registered and unregistered users are able to use this functionality. Type in the tag names, then the photos associated with the tag will be displayed.

Recommendations:**Friend Recommendation:**

- This feature is accessible via the “Friend Recommendation” hyperlink. We use a “friend of a friend” approach. The order of displayed friends is based on their ranks.