Lance Galletti & Wensi You CS660 PA1

**Introduction**: This is a web-based photo sharing application with a database system. It consists of a relational schema (photoshare.sql), the control file (web app.py), and a set of html files.

*The following interaction with the database system are implemented:*

**1 User Management:**

**1.1Becoming a registered user.** A user can register by providing their first name, last name, email address, date of birth, gender and a password. If the user already exists in the database with the same email address an error message is produced informing that the account has already existed.

**1.2 Adding and Listing Friends.**A user can add a new friend on the friend list. When one user adds another user to his/her friend list, the user is also added to the friend list of the other user. The user can search for other users in the system (in order to find friends to add.) Finally, users can view the friends of other users.

**1.3 User activity.** To motivate users in using the site we identify the ones who make the largest contribution and list them on the site as “top users”. 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 are reported.

**1.4 Visitor activity**. As a visitor (not sign-up, or signed-up but not logged-in), he/she can see a sample of photos on the homepage that have been uploaded by other users to this website. A visitor can click on any of these photos and find who owns a particular photo, what album it comes from, who likes this photo, and all comments associated with this photo. A visitor can also click on the owner name of this photo and be transferred into the profile page of the photo owner, and see the albums and friends of this owner. A visitor can also view the other photos in the album that contains this photo. A visitor can click on the “search” button on the homepage, and be transferred to the search page, where the visitor can search for users, comments, or photos. A visitor can also comment on photos but his name will appear as “anon” and will not be a clickable link as this is not a valid user.

**2. Album and photo management**

**2.1 Photo and album browsing**. Every visitor to the site, registered or not, can browse photos. In this project we assume that all photos and albums are made public by their authors.

**2.2 Photo and album creating.** After registration, users can start creating albums and uploading photos. Users are able to delete both albums and photos. If a non-empty album is deleted its photos will also be purged. Users are only allowed to modify and delete albums and photos which they own.

**3.Tag management**

**3.1 Viewing photos by tag name.** Tags provide a way to categorize photos and each photo can have any number of tags. We present tags as hyperlinks which are generated in comments or captions using the ‘#’ symbol followed by at most 40 characters. When a tag is clicked the photos tagged with it are listed.

**3.2 Viewing the most popular tags.** The system can list the most popular tags, i.e., the tags that are associated with the most photos. Again, tags are clickable so that when a user clicks one of them all photos tagged with this tag are listed. Once a user clicks one tag, he/she enters the tag page where they can also see the most popular tags button, and by clicking it they can view the most popular tags.

**3.3 Photo search.** Both visitors and registered users can search through the photos by specifying conjunctive tag queries. For example, a visitor could enter the words "friends boston" in a text field, press the “enter” on the keyboard and be presented with all photos that contain both the tag "friends" and the tag "boston". Users can use the ‘#’ symbol (i.e. ‘#friends #boston’) in the search and the results will be the same.

**3.4 Viewing a user’s own tags.** Under the my tags option of the menu a user can view his/her tags - i.e the set of tags that are used on their photos. When clicking on the tags the user will see the subset of his/her photos that are tagged with this tag.

**4 Comments**

**4.1 Leaving comments.** Both registered and anonymous users can leave comments by clicking a particular photo and inputting their comment in the comment block. Users cannot leave comments for their own photos. If a registered user leaves a comment, this counts towards their contribution score as described above. Visitor can also leave a comment as “anon”.

**4.2 Like functionality.** If a user likes a photo, he/she can add a like to the photo. One is able to see how many likes a photo has and the users that liked this photo.

**4.3 Search on comments.** 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 returns these users ordered by the number of comments that match the query for each such user in descending order.

**5 Recommendations**

**5.1 Friend recommendation.** The system recommends new friends to a user A by finding all friends of A and finding their common friends. The system orders the recommendations based on how many times each recommended friend appears in the lists of friends of friends. This recommendation appears when viewing your own friend list

**5.2 'You-may-also-like' functionality.** Given the type of photos uploaded by a user the system makes some recommendations to them about other photos they may like. The system takes the five most commonly used tags among the user's photos. Perform a disjunctive search through all the photos for these five tags. A photo that contains all five tags should be ranked higher than one that contained four of the tags and so on. Between two photos that contain the same number of matched tags prefer the one that is more concise, i.e., the one that has fewer tags over all.

**6 Additional Functions implemented**

**6.1 Made an online version of this application.** visit at <https://photoshare-app.herokuapp.com>

**6.2 User search.** The system allows the user or visitor to search for user name by entering the search page, inputting the key words either of the first name or the full name, choose the “user” choice on the drop-down button, and press “enter” on the keyboard. On the same page, they can also see the list of all users. So if they want to find somebody and don’t remember the name, they can click the “all users” button.

**6.3 Different view for Users and Visitors.** Both users and visitors can view other users’ profile. When logged in, a user can see a “Friend” button on the profile to allow them to add that user as friend.

**6.4 Unfriend.**  Users can unfriend any of their friends. After doing this they would not appear in each other’s friend list.

**6.5 Allow visitors to leave comment.** Visitors can also leave comments on photos, and their name would be shown as “anon”.

**6.6 Easy tagging.** One can tag a photo by simply using “#” in their comment on the photo.

**6.7 Easy finding all photos associated with a tag**. By clicking on any comment starting with “#”, one can see all photos associated with the tag after the “#”.

**6.8 Multiple ways of uploading photos to make it more convenient.** If the user wants to upload to an existing album, he/she can click on the “MYALBUMS” and then click on the album he/she want to upload into and then click “add photo” button. He/she can also create a new button on the “MYALBUM” page. He can also click the “Creating Albums” under the menu and create an album and upload photos there.

**6.9 A simple “Menu” button to make the page more beautiful and simple**. A menu button is put on the right top corner of each page and by hovering over it the user can see all choices such as “create albums”, “search”, etc.

**6.10 A convenient go-back to one’s profile button.** On the left top corner of each page, a user (not a visitor) can see his/her name, so he can click it whenever he/she wants to return to his/her profile page.

**6.11 View all of a user’s photos.** When visiting a users profile you can see all the photos they uploaded in order of most recent first.

**6.12 Allow users to create an album with the same name.** The system allows users to create albums with same name but is still able to distinguish between them.

**6.13 Allow users to name photos with same name** and the app is able to distinguish between them.

**About the data**

Users

Each user is identified by a unique user id and has the following attributes: first name, last name, email, date of birth, hometown, gender, and password. A user can have a number of Albums.

Friends

Each user can have any number of friends.

Albums

Each album is identified by a unique album id and has the following attributes: name, owner (user) id, and date of creation. Each album can contain a number of photos.

Photos

Each photo is identified by a unique photo id and must belong to an album. Each photo has the following attributes: caption and data. The 'data' field should contain the actual binary representation of the uploaded image file. Alternatively, the 'data' field can store the file location of the file that stores the image. Each photo can only be stored in one album and is associated with zero, one, or more tags. Tags

Each tag is described by a single word. Many photos can be tagged with the same tag. For the purpose of this project we will assume that all tags are lower-cased and contain no spaces. For example, you can have many photos tagged with the tag "boston" in different albums.

Comments

Each comment is identified by a unique comment id and has the following attributes: text (i.e., the actual comment), the comment's owner (a user) and the date the comment was left.