**User management**

**Becoming a registered user.** Before being able to upload photos a user should register by providing their first name, last name, email address, date of birth, and a password. If the user already exists in the database with the same email address an error message should be produced. The other additional information about each user is optional.

**RegisterPage – linkto {IndexPage}**

**Adding and Listing Friends.**You should allow a user to add a new friend on the friend list. For simplicity, you do not have to verify the friendship relationship. Also, you should allow the user to search for other users in the system (in order to find friends to add.) Finally, you must allow a user to list his/her friends.

**Add/SearchFriendPage - linkto{index}**

**IndexPage(PopUp for adding friends, “Cursor”)**

**User activity.** To motivate users in using the site we'd like to identify the ones who make the largest contribution and list them on the site. We'll 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.

**TopContributePage – linkto {index}**

**Album and photo management**

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

**IndexPage - linkto{TopContributePage, FriendPage, RegisterPage}**

**Photo and album creating.** After registration, users can start creating albums and uploading photos. The relevant fields are described above. Users should also be able to delete both albums and photos. If an non-empty album is deleted its photos should also be purged. Users should only be allowed to modify and delete albums and photos which they own.

**Photopage –linkto {UploadPhoto, CreateAlbum}**

**UploadPhotoPage**

**CreateAlbumPage**

**Tag management**

**Viewing your photos by tag name.** Tags provide a way to categorize photos and each photo can have any number of tags. You may think of them as virtual albums. For example, suppose that a user has two distinct albums each of which contains a photo with the tag 'friends'. The means should be provided to view the virtual album created if we select all the photos owned by the user that contain the tag 'friends'. One possible user interface design for this functionality is to present tags as hyperlinks. When a tag is clicked the photos tagged with it are listed.

**Viewing all photos by tag name.** Furthermore, the system should allow users to view all photos that contain a certain tag, i.e., not only the ones they have uploaded but also photos that belong to other users. This functionality is similar to the one above and hence they could be presented together. One possible representation is a switch of the form "View all photos / View my photos" that when toggled switches from one view to the other.

**Viewing the most popular tags.** A function should be provided that lists the most popular tags, i.e., the tags that are associated with the most photos. Again, tags should be clickable so that when a user clicks one of them all photos tagged with this tag are listed.

**Photo search.** The functionality should be provided so that 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, click the search button and be presented with all photos that contain both the tag "friends" and the tag "boston".

**Comments**

**Leaving comments.** Both registered and anonymous users can leave comments. Users cannot leave comments for their own photos. If registered user leaves a comment then this counts towards their contribution score as described above.

**Like functionality.** We want to add a **Like** functionality. If a user likes a photo, should be able to add a like to the photo. Also, we must be able to see how many likes a photo has and the users that liked this photo.

**Search on comments.**In this feature, you want to implement a search function based on the 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 should return these users ordered by the number of comments that match the query for each such user in descending order.

**Recommendations**

**Friend recommendation.**We want to recommend possible **new** friends to a user. One simple approach to recommend new friends to a user A is to find all friends of A and find their common friends. That approach is called “friend-of-friends” and is used in some systems as a first step to make friend recommendation.  Order the recommendations based on how many times each reccomended friend appears in the lists of friends of friends.

**'You-may-also-like' functionality.** Given the type of photos uploaded by a user we'd like to make some recommendations to them about other photos they may like. To achieve that do the following: Take 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.