CMPSCI 445: Information Systems
Fall 2008
Project Description

Last updated: Sep 29th

1 Overview

In this course project you will design a web-based online social networking application. Your site should provide an interface that allows users to register, search for friends, post status messages, as well as provide some services for the administrator of the social community. Many of you will already have had personal experience using social networking sites like Myspace, Friendster, or Facebook. You should use these sites as inspiration in the development of your site since many of the project requirements are based on common social networking features. You may also want to read more about online social networks here: `http://en.wikipedia.org/wiki/Social_networking`

The online social networking site you design and develop needs to (i) satisfy minimum functionality requirements, (ii) provide good performance for common user operations, and (iii) implement selected advanced features in addition to the basic functionalities. The grading of the project combines all three requirements using the scheme shown below:

60% Correct implementation of minimum functionality

20% Performance of required features

20% Advanced features

2 Minimum functionality

Your application must support the following basic features.

2.1 Registration

Access to your application should begin with a registration page allowing new users to create a profile or allowing existing users to login. Users may need to edit their profile more than once after creating it the first time.

2.2 User profile

Each user will have a profile page describing basic information (name, username, personal interests, hometown, date of birth, gender etc.) along with a list of their friends. Only registered users should be able to view profile pages, and once logged-in, a user should be able to view only their own full profile and the full profile of their friends. If an attempt is made to view the profile of a non-friend, an appropriate response should be presented, with the option to add the user as a friend. The profile page should also include a message list, displaying messages recently sent to the user by others, with the most recent messages listed first.

2.3 Search

Your application should allow users to search by name and should return both exact matches and substring matches (using LIKE). Your application should also include a general search that will match the query string in any field of a user's record.

2.4 Friendships

Users should be able to create friendship relationships with other users whose names are returned in search results or who are visible as friends of friends. Once requested, friendship relationships will be considered automatically approved (unlike most real social networking sites) and should be stored by the system. Users should also be able to remove any of their existing friends.

2.5 Messaging

Your application should include the ability for users to "send" messages to their friends. Once a message is sent it should appear on the friend's profile page.

2.6 Administrative Interface

Your application should have a page, accessed only by the *admin* user, which should report summary statistics on the community (e.g. number of users, number of friendships, the top ten most "popular" users, gender ratio in the population, etc.)

3 Performance

Your application is expected to run with reasonable performance. While there are no specific requirements in terms of response time, most operations on the web site, including user registration, user login, searches, and updates, should be completed with no obvious delay.

If you observe significant delay in operations in your application, it is important that you review your design and implementation to identify the causes of the slow performance. As a general guideline, you may want to consider the following things:

- Can you create more indexes to improve the performance? The class material relevant to this task is physical design and tuning. The relevant SQL command is CREATE INDEX index_name ON table_name (column_name(s)).

- Did you implement the functionalities of your application in the right places of your program? For example, a join between two tables should be performed in the database backend. Performing the join in your PHP code is a mistake – even though you may still provide the correct answers, the performance of your application will suffer.

- Can you refine the database schema to improve the performance?

4 Advanced Features

Your application is expected to have at least one advanced feature beyond the basic functionalities described above. The following list presents several ideas regarding the project enhancement.

1. Administrative Log – Enhance the administrative page so that it displays a log of all recent actions taken by users. This should include users' changes to their profile, friendship additions and deletions, etc.

2. Flexible schema – Allow users to freely add descriptive fields to their profile. For example, if a user wants to add *political affiliation* to their profile description then they should be able to add this field and an appropriate value. Display all such fields in the user's profile, and also include these fields in the search functionality.

3. Friend recommendations – Add features that can intelligently recommend potential friends for users. You should explain the basis of your recommendations and try to rank them by a reasonable metric. You may consider recommendations based on common friends, or similarity of a users' descriptive fields, or a combination of the two.

4. Security – Implement one or more of the following security features: secure use of cookies, resistance to SQL injection attacks, secure storage of passwords, secure urls.

5. Scale – Explore the implications of scaling your database by a factor of 10.

You are also welcome to propose and develop new features not listed above. Just be creative and have fun!

5 Sample Data

You will be provided with a sample data set with basic descriptive fields for 10,000 users. You should issue a bulk update operation to populate the database with friendship relationships. Ultimately, you should give each user approximately 100 friends, so that you end up storing about 1,000,000 friendships. You may create additional users and friends manually in the course of using and testing your application, but in the end you must have at least this number of users and friendship relationships.

6 Deliverables

You are required to turn-in the following (please see the project page of the website, or the calendar for due dates).

1. Proposal - Submit a description of your planned application. Describe the type of content your site will handle, how you will meet the minimum requirements, and which extensions you plant to implement (along with any special issues the extensions may involve). You should also include an E/R diagram or relational schema for the data to be stored in the database supporting your application. Finally, break you project into small tasks and include a timeline with the dates by which you intend to complete those tasks, and, where appropriate, which group members will be responsible for those tasks.

2. Midterm Assessment - Submit an overview of your progress and a link to your partially-completed website. Update your timeline from above indicating which tasks have been completed to date, and refine the remaining tasks to be completed.

3. Final Presentation & Report - Final presentations will happen in-class during the final two course meetings. Groups will be assigned one of these dates randomly. A live demonstration, and description of your implementation by the group will be required. A final written report will be due by the end of finals week. The written report must include a copy of all code, and full description of your extensions.

4. Individual contributions – Each group participant must individually report by email on their contributions to the group project. Describe the tasks to which you contributed.