CPSC 240

Group Project Description:

Create a social network service that enables users to send and read text-based messages. Message length should be limited to 140 characters.

It's up to you exactly how amazing and cool you want your messaging service to be. It could be a pretty simple, bare bones, and functional application. Or, it could be a killer app that looks really slick and has a lot of extra features. More ambitious efforts will be rewarded with higher grades, but there is certainly nothing wrong with creating a modest, working product.

In order to get a low 'B' on this project your site must implement the following features:

- Users can select to register with the service. Registered users can opt to create a profile of information. This profile will be publically accessible by default. Registered users may post and view messages. Users can also choose to view public messages on the site without registering.

- Users can subscribe to other user's messages. When posting a message, the poster may choose to make the message publically viewable, or may choose to send it only to his/her subscribers.

- When a registered user logs in, he/she should see a list of the messages posted by the people they have subscribed to.

- Information posted to the service must be persistent. If the system is shut down, all user information must be available exactly as before when the system comes back online.

In order to get above a low B this project, you must implement all of the features in the 'B' group and implement the following features:

- If desired, a group of users, can post together on a topic by using a hashtag, a word or phrase prefixed with a '#' character. Posts should be searchable by hashtag.

- Users can respond to a message by using the '@' character followed by a username.

To move into the 'A' range, your group must add additional significant features. You do not have to implement all of these. The more you implement, the higher it pushes the final grade.

You may invent your own addition, *or* choose from the following list of ideas:

- Create a graphical user interface for your service
- Use a real database (not just a text file) to store data on the backend of your service
- Add statistical tracking of phrases so that users can discover trends in posted information. Track the leading trending topics and display them in your user interface.
- Ability to upload a photo (as an image file) and post the image in a message on the service.
- Create a networked service so that users can connect to your Twitter server, register their account and send/receive tweets live in a distributed fashion.

Important note

The above feature sets are *examples*. You do not have to make your application work in exactly those ways. You are free to be creative in almost any direction you see fit! I will judge your final product by estimating the difficulty (and awesomeness) of your features and comparing them to the above benchmarks. If you have any questions about what a particular feature would be "worth," feel free to explain it to me and ask before you begin implementation.

Due Dates

The final code for the project will be due: Mon April 18, in class. Starting that day, groups will demo your final system to the class.

There will be additional deliverables related to the project including (but potentially not limited to) a requirements/system design document. This document will be our 3rd and final writing assignment for the course.

Time is limited and speed and efficiency is essential for project success. I recommend spending about a week making critical decisions about project scope, included features and how they will work. This is also a great time to discuss version control software that will be used by the team and to make decisions related to software process. After about a week, you'll need to be working on the implementation and related assignments.