**CS 497/597: Programming Assignment 1**

**SOCIAL Network**

**250 points**

**Due date: 09/23/2014**

Course: Cloud Computing

Term: Fall 2014

Instructor: Vijay Dialani

# Introduction

Social networks such as Facebook, LinkedIn, Twitter and Google+ allow a community of users to connect and interact by exchanging messages amongst them. These messages can refer to other users using ‘@’ prefixed to the username or they could be referencing a hashtag topic prefixed by a ‘#’. Each user could reference a group of users using ‘@@’ sign that references a group. Recipients of the message need not be online to receive the message, messages held for them at the server and are delivered at subsequent login. Most of these social networks provide an API to access the contents, one such example of Twitter API can be found at <https://dev.twitter.com/docs/api/1.1>

# Social Network Service

As a part of this assignment two web service interfaces to the social network needs to be designed and implemented (NOTE: web services interface is a must, no scripting please!). The system should allow for creation of new user account, using their username/email and a password (no need to implement the password reset functionality). Once a user has logged in they can search and follow other users in their network by sending a request. Note: Friend requests are treated as system level messages.

The **Friends and Follower** service should support the following RESTful API calls

|  |  |
| --- | --- |
| **Resource** | **Description** |
| GET friendships/incoming | Returns a collection of numeric IDs for every user who has a pending request to follow the authenticating user. |
| GET friendships/outgoing | Returns a collection of numeric IDs for every protected user for whom the authenticating user has a pending follow request. |
| POST friendships/create | Allows the authenticating users to follow the user specified in the ID parameter. Returns the befriended users id when successful. Returns a string describing the failure condition when unsuccessful. If you are already friends with the user a HTTP 403 may be returned |
| POST friendships/destroy | Allows the authenticating user to unfollow the user specified in the ID parameter. Returns the unfollowed user’s id when successful. Returns a string describing the failure condition when unsuccessful. |
| GET friends/list | Returns a cursored collection of user ids for every user the specified user is following. |
| GET followers/list | Returns a cursored collection of user objects for users following the specified user. |

The **Tweet** Service should support the following RESTful API calls

| **Resource** | **Description** |
| --- | --- |
| POST tweet/tweet/:msg | POSTS a new tweet, 128 characters max and returns a unique tweet id for this message. |
| GET tweet/show/:id | Returns a single Tweet, specified by the id parameter. |
| POST tweet/destroy/:id | Destroys the status specified by the required ID parameter. The authenticating user must be the author of the specified status. Returns the destroyed status if successful. |
| POST tweet/retweet/:id | Retweets a tweet. Returns the original tweet with retweet users id embedded. |

# Social Network Client

Develop a simple web client using your language/framework of choice to demonstrate the functionality of your web service. The client should support the set of function calls mentioned earlier. The name of the web client should be *Chatter* and it should be packages as a war for deployment in apache tomcat 8.0 or higher.

# Extra Credit (20 points)

* Develop a Web Client that uses CSS and whose appearance could be easily modified.

OR

* Demo your project and design in the class

# Submitting the Assignment

* There must be a README that mentions the name of the student, course number, and assignment number. It should provide details about any assumptions made, configurations and comments.
* The assignment must have a pom file that generates war files that can be deployed in Apache tomcat server.
* Assuming all your assignment files are in your directory ~/cs597/p1 on onyx, change the directory to p1 and type the following command.

submit vijaydialani cs597 p1

or

submit vijaydialani cs497 p1

The submit command will provide you with a timestamp and directory path name to confirm your submission.