# CSE3310 Project

The Social Network

# Iteration II

## Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| REQID | F/NF | Requirements | Test Steps | Test Status |
| 1 | F | The list user option shall list all the users in an orderly fashion. |  | Pass/Fail |
| 2 | F | The show users option shall show the specific user and the posts they have made. |  |  |
| 3 | F | The edit option shall let the user edit the information entered, by default it is saved to a disk file and read in on program start. |  |  |
| 4 | F | The resync option shall forget all saved data and reset the sending of data. |  |  |
| 5 | F | The post option should let user post their content. |  |  |
| 6 | F | On start the program should print user’s personal information which will be published not more than once per 30 seconds. |  |  |
| 7 | F | After receiving request, data will be sent out 1 per minute, which contains vector of UUID and serial numbers. |  |  |
| 8 | F | The serial number of the post will be the unique identifier for the post. |  |  |
| 9 | F | In order to make a request there must be nodes online with data you do not have. |  |  |
| 10 | F | The program shall ask for the user information only for the first time. |  |  |
| 11 | F | A user must be able to post, exit, and rejoin at any time they want. |  |  |
| 12 | F | Posts are to be sent only when requested to save network traffic. |  |  |
| 13 | NF | User id must be of length 37 generated by Boost stored in a char array |  |  |
| 14 | F | All post must be stored on local disk |  |  |
| 15 | F | The Stats option shall print out how many nodes are known and how much content is available in this node listed as a percentage |  |  |
| 16 | F | Local disk file must be encrypted |  |  |
| 17 | F | The menu will display the user commands. |  |  |
| 18 | F | All serial numbers generated will start at 0 then generate new one every time post is made |  |  |
| 19 | F | UUID generated will be stored in a file. |  |  |
| 20 | NF | This program should be compiled and run in Linux OS. |  |  |
| 21 | F | An application will only satisfy requests for data it owns. |  |  |
| 22 | F | Program should support command to list all the users |  |  |
| 23 | F | Program should support command to reset the sending of data. (resync) |  |  |
| 24 | NF | TSN will be coded in C++11 Language |  |  |
| 25 | NF | The IDL for the program must be IDL provided by the instructor. |  |  |
| 26 | F | Notification should be displayed if new post is made or user is online. |  |  |
| 27 | F | The posts will be matched based on the interests the user selects. |  |  |
| 28 | F | The program shall support Direct messaging. |  |  |
| 29 | F | The program should be built upon the functionality provided in iteration I. |  |  |
| 30 | F | The user Interface shall be fun and user friendly to use. |  |  |
| 31 | F | The direct message shall be displayed as soon as it arrives. |  |  |

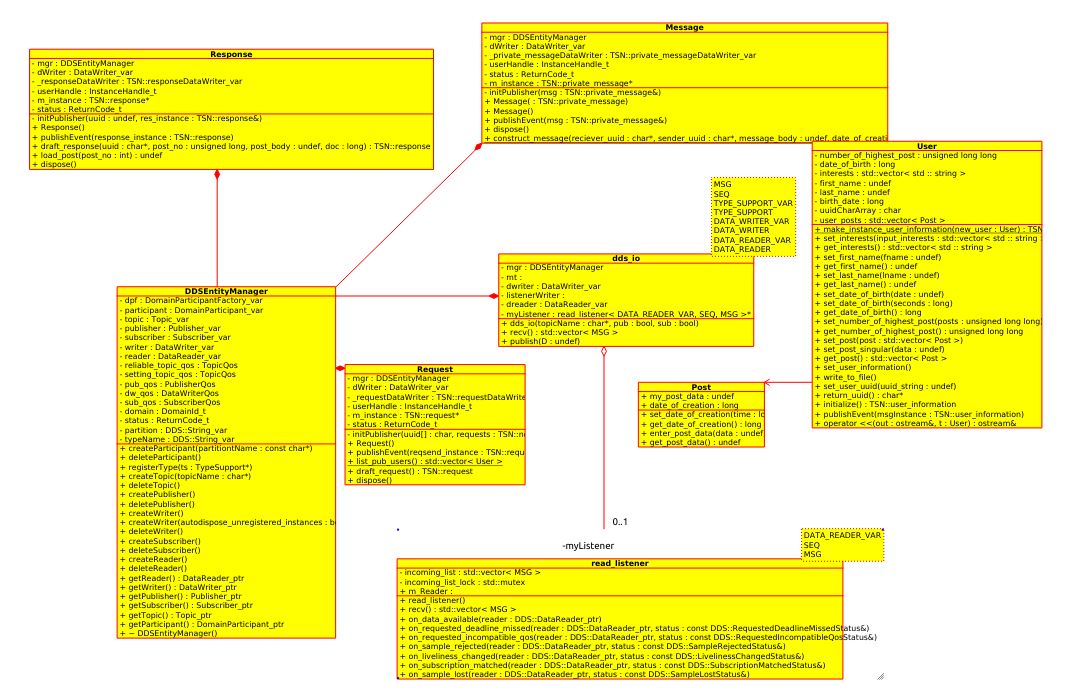
Tabular Use-Case for user Interface:

|  |  |
| --- | --- |
| **USER ACTION** | **SYSTEM RESPONSE** |
| List User | Lists all the user in orderly fashion |
| Show User | Show specific user and post they have made |
| Edit | Ability to edit personal information |
| Resync | Deletes all the user info and start from the beginning |
| Post | Gives ability to post |
| Show Statistics | Shows the number of times post is made |
| Send message | Send Private message to selected user |
| Exit | Exits out of the program |

## Design (4 points)

Provide:

* A class diagram.



XMI of the class diagram is included in the tar ball.

* Overall description of the program and how it works (text).

TITLE: The Social Network

**Iteration II**

The second iteration of the project will add more functionality and easy user interface in the program. The two primary functionalities are user being able to send private message to another user on the network and getting notification if any user is online. This version of TSN is the modification on the previous version which is better and easy to use.

Based on the class diagram and nature of the program, Once the program executes in the command line interface and the user has never signed in before several prompts for personal information will show up, where the name of the users will be stored in the first\_name, last\_name variables. Similarly the date of birth will be stored in birth\_date as long and the interest will be stored in a vector of string called interests which are inside the User class as shown in the UML diagram above. As soon as these information are received the user gets a ID which is stored in uuidCharArray and if any posts are made by the user the number of posts are saved in the variable number\_of\_highest \_post, and the post itself is stored in a vector of post called user\_posts which are saved in a file locally. In the same User class we have several getter functions. Similarly, In the class Post, we have my\_post\_data to save the post initially before saving to a file, also the variable date\_of\_creation saves the time and date of the post made and lastly we have some setter and getter functions in this class.

**Testing:**

The system testing of the project is given below:

Steps to compile and run the project:

1. Open command prompt and go to open splice directory, do source ./release.com
2. Go to the program directory and locate make file, do make and ./main
3. User interface will show up, and testing begins

We conducted the test in two computers both running on Linux OS and on same network.

After the compilation of the code on both the computers, we executed the code. The program prompted for personal information on both computers. After entering the personal information on both computers, we received UUID and user number for both users.

After the generation of UUID, the user options were tested

1. **List Users**- When user 1 enters this option, the list of users on the network will show up. In this case name of user1 and user2.
2. **Show User**- When user 1 enters this option, the program will prompt for user number, after user number is entered the posts and information about that user number is shown.
3. **Edit-** When user 1 enters this option, the program will let user 1 edit information entered by user 1.
4. **Resync-**
5. **Post-** When user 1 enters this option, the program will prompt for the post that user 1 wants to make, which is then saved in a file locally.
6. **Show statistics-**
7. **Request Post-**
8. **Send message-**
9. **Exit-** This option will exit the user out of the program.