Walter Matthews
COMP 2710
Alvin Lim
10/04/2017

Lab 1 Design Portion

Analysis

Use Cases: This program is designed to display a message sharing application with a text-based interface. Users will be introduced to a few commands upon start up. **Create a new user** will create a user give the user access to the features available to the program. **Broadcast a message** will allow the user to send a message to any user in the social network. **Multicast a message** will allow the user to send a message to a particular group. **Unicast a message** will allow the user to send a message to one other user. **Display wall page** displays the sent messages from the user that you are currently logged in as. The messages will be displayed in reverse chronological order. **Display home page** will display all messages received by the currently logged in user. **Create a group** creates a new group where multicast messages can be used. **Join a group** adds the current user to any existing group in the network.

Switch to different user allows the user to change between users already in the system. Finally, **Quit** will end the program.

Design

The implementation will have multiple classes each designed for a specific purpose. These classes include:

Menu Class: handles when a user does something the program will not allow without actually messing up how the program runs

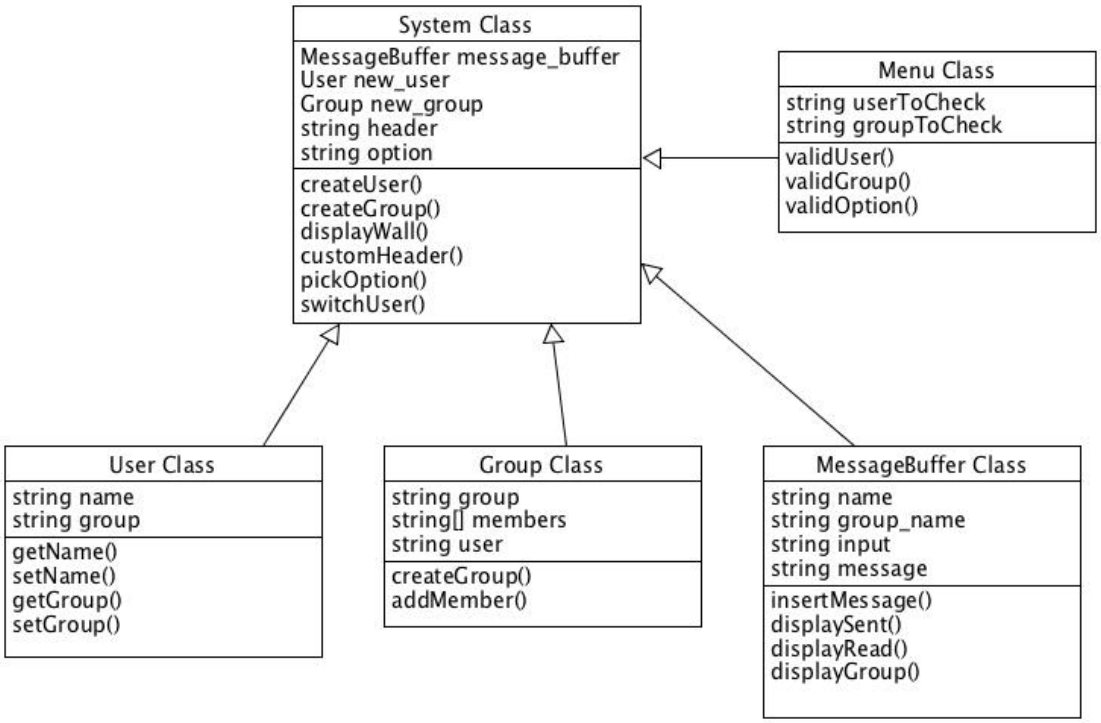
System Class: Instantiates objects that must be initialized

User Class: Maintains all information on the user names, group names, and wall pages.

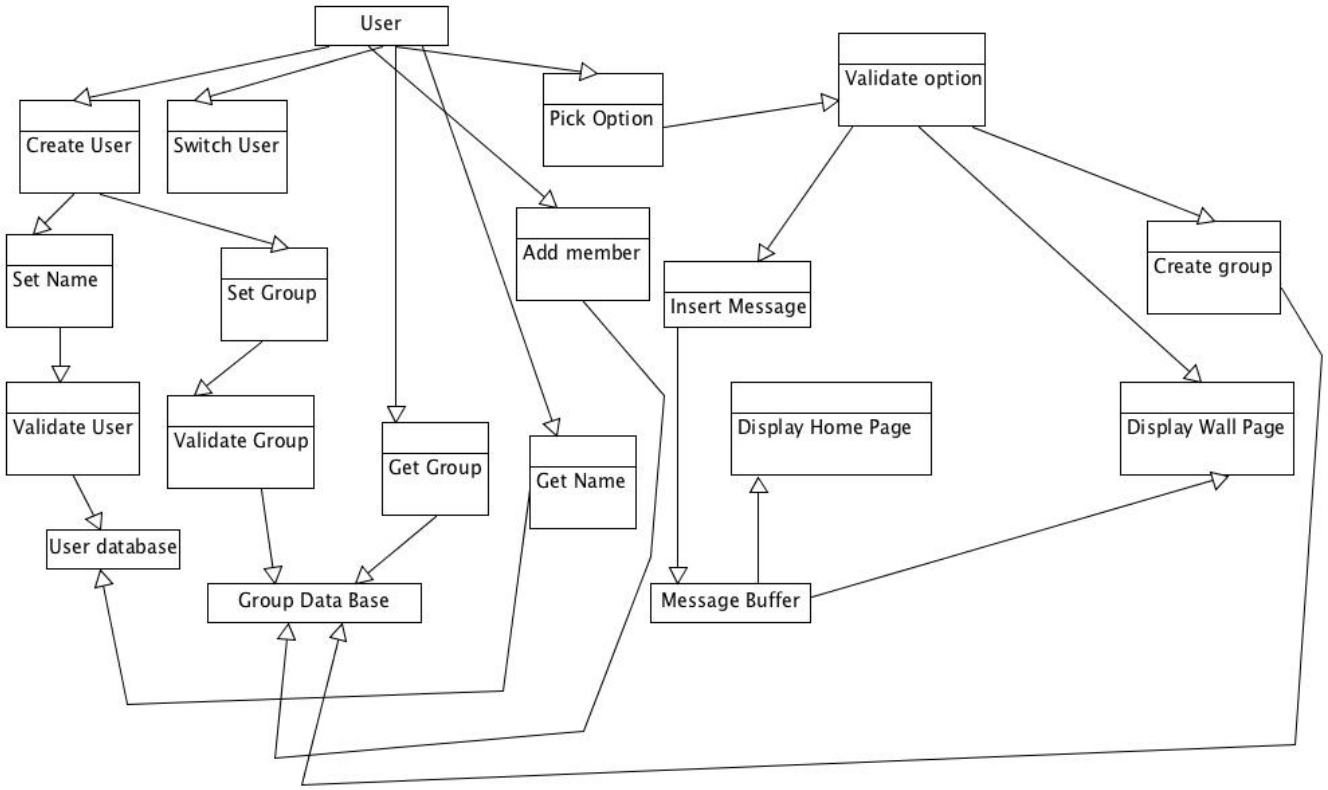
Group Class: maintains the members of the group

MessageBuffer Class: maintains the message buffer used to store all messages.

Class Diagram



Data Flow Diagram



Testing

System Testing:

Create User -> Enter username

Create User -> Enter existing username

Create User -> Enter empty String

Join group -> enter existing group

Join group -> enter non existing group

Join group -> Enter group without #

Join group -> Enter empty string

Create group -> enter group name

Create group -> enter existing group name

Create group -> enter empty string

Create group -> enter group without #

Broadcast -> enter message

Broadcast -> enter empty string

Broadcast -> enter large message

Broadcast -> find message on every page

Multicast -> enter message with existing group

Multicast -> enter message with non existing group

Multicast -> empty string

Multicast -> find message on group members page

Unicast -> enter message with existing user

Unicast -> enter message with non exisisting user

Unicast -> empty string

Unicast -> find message on receivers page

Display Wall Page -> look at wall page of existing user

Display Wall -> display wall of user with no sent messages

Display Home Page -> display wall of current user

Display Home -> display wall of user with no received messages

Switch User -> switch to existing user

Switch user -> switch to non-existing user

Switch user -> empty string

Quit -> quit, receive goodbye message

Unit Testing

getName() -> call when current_user is defined

getName() -> call when current_user isn't defined

setName() -> regular string

setName() -> long string

setName() -> empty string

getGroup() -> call existing group

getGroup() -> call non existing group

addMember() -> add current_user

addMember() -> call when current_user is undefined

validUser() -> insert string

validUser() -> insert non string

validUser() -> insert long string

validGroup() -> call existing group

validGroup() -> call non existing group

validGroup() -> call group without #