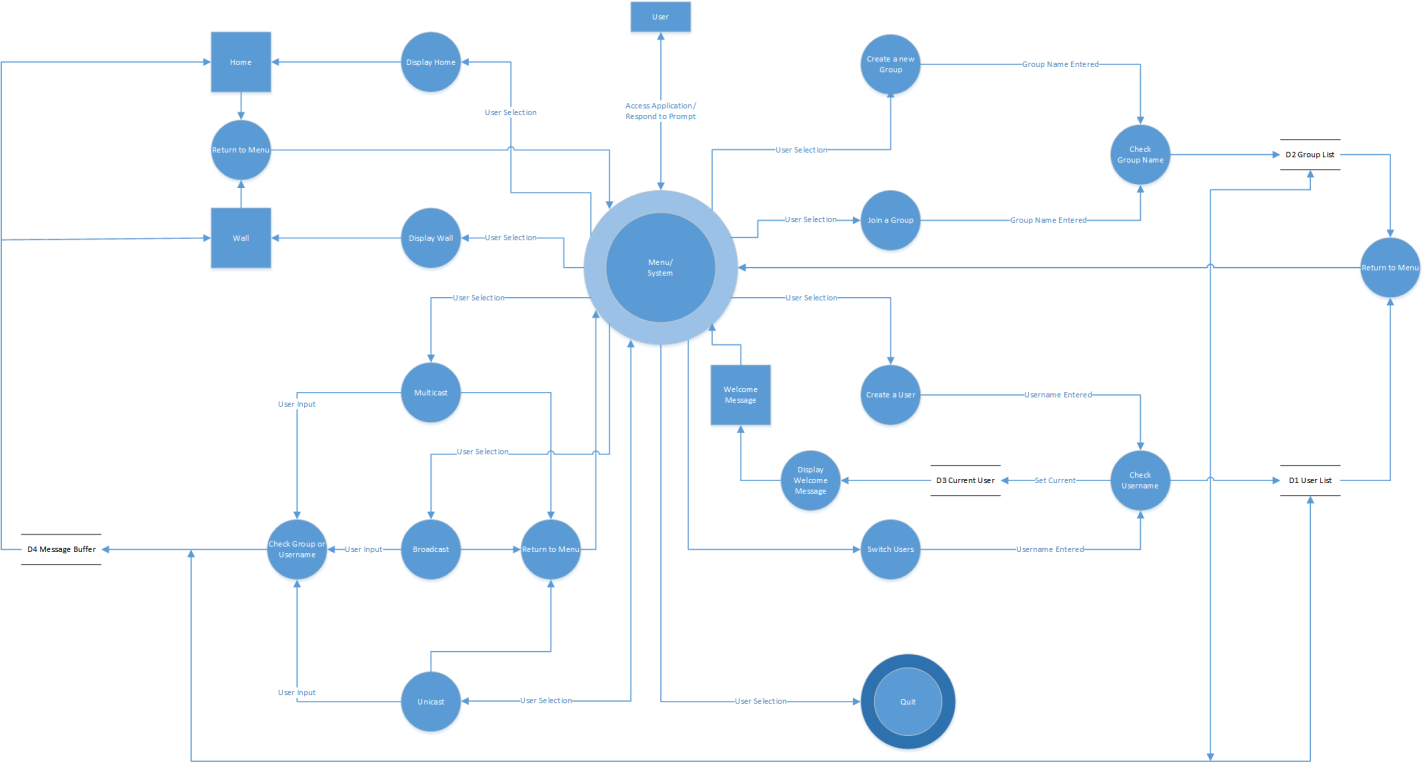
Brandon Hurler Software Construction 9/17/2014

# Analysis

1. Create a new user
   1. Ask the name of the new user, makes sure the user name is not already in use, add the user to the “All” group, and then welcome the new user.
   2. Newly entered users become the current user for the session unless otherwise changed.
2. Broadcast a message
   1. Allows the user to input and send a message to all other users.
3. Multicast a message
   1. User can input a valid group name to multicasta message to, then input and send the message.
4. Unicast a message
   1. User can input another user’s name and then input a message which will be sent to that particular user.
5. Display Wall Page
   1. Will display the user’s sent messages with the group name (multicast), recipient’s name (unicast) or “All” (broadcast) in reverse chronological order.
6. Display Home Page
   1. Displays *User*’s Home Page and the latest 2 messages of any type, prompting the user if they want to see more.
      1. If no, display no more; if yes, display all the rest.
      2. If 2 or fewer, do not prompt for more.
      3. All displayed in reverse chronological order.
   2. Show all messages that the user sent and saw.
7. Create a new group
   1. Prompt for the name of the new group that is being created, make sure the group name is not already in use, and then create the group. If the name is already taken, prompt for another name and repeat.
8. Join a group
   1. Ask what group the user would like to join, check for the group, check that the current user is not already in the group, and then add the user to the group if it exists. If the group does not exist, ask for the group name again and repeat. If the current user is already in the group, display a message informing the user they are already a member.
9. Switch to a different user
   1. Ask which user that will be switched to, check if that user name exists, make that user the current user, and welcome the current user back.
10. Quit Auburn Messaging System
    1. Post a thank you message, and then end the program

# Design

* Classes
  + Menu – manages user inputs and their choice, along with any associated errors.
    - Functions
      * CreateUser
        + Calls functions from the User Class to create a new user.
      * CheckUser
        + Calls functions from the User Class to validate the username.
      * SwitchUser
        + Calls functions from the User Class to switch currentUser.
      * SendMessage
        + Calls functions from the Message Class to add one of the three types to the message buffer.
      * CreateGroup
        + Call functions from the Group Class to create a new group.
      * JoinGroup
        + Call functions from the Group Class to join an existing group.
      * Error
        + Handles user errors.
      * Quit
        + Exits the program.
    - Variables
      * string currentUser // May exist here or in the User class. Unsure now.
  + User – manages user information, such as: username, friends list, wall page, etc.
    - Functions
      * Home – manages user’s home.
      * Wall – manages user’s wall.
      * UserList – builds and stores usernames.
    - Variables
      * string currentUser // May exist here or in the Menu class. Unsure now.
      * string users
  + Group – maintains all groups and their members.
    - Functions
      * GroupList – builds and stores group names.
    - Variables
      * string groups
  + Message – manages the message buffer; insert, read sender, group recipient, messages, etc.
    - Functions
      * MessageBuffer – builds and stores the messages from unicast, multicast, and broadcast.
      * Unicast – builds the unicast messages.
      * Multicast – builds the multicast messages.
      * Broadcast – builds the broadcast messages.
    - Variables
      * string messageBuffer
      * string unicast
      * string multicast
      * string broadcast

# Testing

1. Create New User
   1. Create the initial user.
      * Make sure Welcome message prints correctly.
   2. Attempt to create a user that already exists.
   3. Create multiple users.
2. Send Message
   1. Attempt to send a message of each type with the current user, when only one user has been created.
   2. Attempt to send a message of each type with the current user after multiple users have been created.
   3. Attempt to send a message type to a user not yet created.
   4. Attempt to send a message of each type to the current user.
   5. Attempt to send a message to a group not yet created.
   6. Attempt to send a message to a valid group.
   7. Attempt to send a message to everyone.
3. Create a Group
   1. Create the initial group.
   2. Attempt to make an already existing group.
   3. Attempt to create multiple groups.
4. Join a Group
   1. Attempt to join an invalid group.
   2. Attempt to join a group the current user is already a member of.
   3. Attempt to join an existing group.
5. Display Wall
6. Test to make sure wall page displays for current user.
7. Test to ensure that the two most recent messages of the user are displayed first.
8. Test to make sure wall page asks the user if more messages should be displayed if more can be displayed.
9. Test if the user opts to display more messages that the rest of the messages for the user are displayed.
10. Test if the user opts not to display more messages that the rest of the messages for the user aren’t displayed.
11. Home Page
    1. Test that it displays the current user home page title.
    2. Test it displays only the two latest messages from either from broadcast, multicast or unicast messages in reverse chronological order.
    3. Test proper formatting.
    4. Test that after displaying the latest two messages, it will then prompt the user if they want more.
       * Test yes and no responses.
    5. Test if there are two or fewer messages that the program will not prompt for more messages.
12. Switch User
    1. Attempt to switch user to a user that has not been created yet.
    2. Attempt to switch user to the current user.
    3. Switch to a user that has already been created.
13. Quit
    1. Make sure that when user selects to exit program, the program terminates and closes.

Overall System Testing

1. Test user input at menu start
   1. Test negative number
   2. Test a word
   3. Test multiple words on one line
   4. Test correct input
2. Test display wall page option
   1. Test after new user has been created
   2. Test after switching user
3. Test display home page option
   1. Test after switching user