**CS-230 Group 22 Report**

Ricky Stephens (830169)

I implemented the following classes:

* Account
  + Hold all data about each account
* BST
  + Binary tree made up of BSTNodes that contain each Account on the system.
  + Allows 4 ways to retrieve accounts from the BST:
    - Beginning with (binary until first match found, then checks every account below for matches, as usernames that begin the same will not always be next to each other in the tree).
    - Contains (checks every account in the tree to see if it contains the search string).
    - Exact match (binary search).
    - Return all accounts.
* BSTNode
  + Multiple nodes make up a binary search tree.
  + Each node contains a reference to an account.
  + Each node contains a reference to a left and right node.
* Graph
  + Made up of objects of the Vertex and Edge types.
  + Contains hashmap of references to all vertices.
  + Contains hashmap of references to all edges.
* Vertex
  + Contains up of a username string, to identify it by.
  + Contains an arraylist of all edges, of type Edge, that this vertex is involved in.
* Edge
  + Contains two references to the vertices, of type Vertex, that the edge involves.
  + Contains a weight value of the edge, of type int.

After getting the basic implementations of these classes done, during the group meetings there was a lot of discussion about the best way to return results from the searches in the BST, returning a list of friends from the graph, etc, and we needed to agree on these in order for the classes to collaborate properly. For example I originally returned search results from the BST as an array of Account objects, however the read/write classes used arraylists, we decided that it would be much easier and more efficient to use an imported arraylist to return account searches as the length of this data structure cannot be too long or too short, and an index isn't required, unlike when using an array. Changes like these were important to allow my BST and Graph to collaborate and be populated properly by the read/write classes. Each group member has made a lot of useful suggestions to other members of the group, for example I suggested a simple way of sending multi-recipient messages using a loop to send the message to each recipient individually using the standard methods for sending messages, rather than doing something more complex like a group conversation. On the whole the project has been successful, any problems being fairly easily overcome due to the helpful suggestions of the other group members.

Thomas Booth (864469)

In this assignment, I implemented the following classes:

* Line
  + This stores all information about a 2D Line object.
* DrawingPanel
  + This extends from JPanel to allow the user to draw on screen using MouseListeners and MouseMotionListeners.
  + This class also saves each line/particle trace as its been drawn to a .txt file. It also has a mini FileReader within it to allow the loading of a .txt file to bring up previous drawing history.
* CollabDrawGUI
  + This extends from the GUI class created by Carl. This GUI displays two draw buttons, which the user can click to decide which draw tool they’d like to use. This GUI also has 5 colour buttons (black, blue, red, green and yellow). The user can click on these to choose which colour they’d like to draw in.

I was also responsible for producing the minutes from each of our group meetings and producing this group report on behalf of the group.

Getting a simple implementation of classes completed and working together was definitely harder than I anticipated, but after meeting with my group and discussing several ways of implementing things, it did make things easier to understand.

Throughout this assignment, I myself have encountered many difficulties with regards to my part of Skypertawe but, with the help from my group members, have been able to overcome many of these difficulties. With regards to the design of Skypertawe, I did not implement a ‘ParticleTrace’ class like specified in the design. This is because as I was developing this part of the system, I felt it would be easier to do things a different way. Even during implementation I changed my mind a couple of times. For instance, I had originally created a LineDraw and TraceDraw classes, each handled the drawing of a line and particle trace respectively, but the problem here was that I had problems with the GUI, so I decided to change to what I have made now.

As previously said, this assignment was harder than first thought, but suggestions made from everyone in the group made it easier to overcome. At first getting the group to meet together was relatively easy as we were all eager to get an early start on a project we knew was going to challenge us, and so with this work ethic, we have worked together well.

Carl Thomas (662735)

For the group assignment, I took the role of Planning and Quality Manager, this role requires me to ensure the quality of work produced by the group was of a high standard and check if the team is following the correct protocols in terms of meeting and whether or not everyone was at an equal level of understanding with the task in hand. Going from design to implementation, we made some changes to our program in the interest of saving time and simplifying the hierarchies, as a group we omitted the VideoMessage, PhotoMessage and their respective GUI classes as it was not necessary to include them in the implementation.

For the implementation process, I contributed the following classes:

* GUI
* MessageGUI
* LoginGUI
* CreateAccountGUI

Despite initial success with creating the GUI and LoginGUI classes, I later had trouble with figuring out ideal layouts for the remaining classes due to my unfamiliarity in the code required for making GUIs. However, after referencing the Java API, I was able to create and implement these classes using what I believe were ideal layouts. Another problem I faced was trying to figure out how to link the GUI classes together in one hierarchy, however other members of the group were willing to assist me in figuring it out and getting it to work.

Wenju Mu (918014)

In the early stage of the project actively, I was involved in the idea of creativity and the overall design of the project. Although I haven’t spoken much throughout, I try my best to follow our members’ ideas.

In coding stages, I am responsible for the preparation of the following procedures. To complete the Message, FileMessage, TextMessage, URLMessage which are help Tom to finished message GUI design. Cooperate with Dan to complete the ReadWrite parts. The function of sending information in many form is realized.

During the group meeting I have questions about Message with MessageGUI. Dan definite what should I do and allow me do backend for files. When I finished my parts members can continue do ReadWrite parts. In every meeting, if I have some parts is not clearly Carl always explain the meaning and teach me about login part. In the whole meeting, everyone can ask and discuss questions and giving a useful answer. I am responsible for the part of the code is completed by the cooperation. So, a clearly comments and accurately identify is necessary. Therefore, I have a short discuss with Dan and we reached a consensus view.