Project Report Part 1

Functionality

Our project was to create a social media profile application that, in general terms, allows users to connect to a server using a client and create their profile and also add other users on the network as friends.

The first feature is obvious but important and it's the ability for our application to be used by multiple users at once. We achieved this by using a mix of multithreading and a temporary connection system where the clients only connect to the server to transmit necessary information. We believe this is similar to how modern web applications function, as a website doesn't set up a continuous connection to the server.

Next, all interactions with users had to be GUI based. This was achieved through use of the Java Swing and AWT libraries in the client.

Another important element to our application is that all data entered in is stored and persistent, regardless of whether the user is still connected or not, or if the server itself has been shutdown or crashed. All the data will still be there for users to access.

Descriptive errors using JOptionPanes have also been implemented in order to inform the user on any issue encountered.

Now for the main features of the program, users can create accounts when using the application and access to the application is only allowed through an account login consisting of a username and password. The accounts can then be edited by the user if he or she chooses to once they are logged into the application. Profile information can also be edited, including interests, the bio, phone number, email address, and more. Profiles can also be deleted at the request of the user who created it. An integral feature that makes this application a "social networking" application is the ability for users with profiles to add other users as friends. There is a friends list window that will pop up once users click on the number of friends they currently have (via JOptionPane). There, the user can see all the incoming and outgoing friend requests that the user has received and sent to other people on the application. The user can also open the profile pages and friend lists of those users in the friend window. From there, the user can then choose to reject or accept the requests and then the request turns into a friend. Accounts can also be unfriended once friended if the user desires. There is also a search feature integrated into the application. Users can search for other accounts on the application and from there, the user can then go and send friend requests to those people they searched. There is also an option to list all the users on the platform in the search menu by just keeping the search bar empty and then just pressing search. The basic menu is always open as clicking on buttons only open up new windows on top of the menu. This is to allow the user to continue using the program and providing ease of access even after the user closes the window of the task they are currently doing. To log off the service, the user has to exit out of the main menu.

Design

In regards to the design of the application, when the user first uses the program, the first thing that appears is a login screen. The user can then choose to either login with an existing account or create a new account to access the application. An account is needed to access the application to ensure security. If the user decides to create a new account, they can enter a username, a password, an email, a phone number, a bio and interests the user has.

After successfully logging in, the user is met with a general menu where the user can choose what they want to do next. This menu should always be open even after clicking on a button on the menu except when exiting the client. One option in the menu is being able to look at and edit your own profile, a common feature in most modern social media and allows the user the flexibility to customize their own profile to their liking whenever they want. From this menu, the user also has the option to delete their profile should they choose. If their profile is deleted, the user will then be met with a confirmation window that asks if the user's choice is final and if yes is pressed, the program is exited after a confirmation message.

The user can also view their friends list from this menu. In the friends menu, the user can see incoming and outgoing friend requests to other accounts on the platform. When opening the friends list, a new window pops up to provide ease of access as once you are done looking at the friends list, the user can just close the friends list and be back at the menu. The profiles and friends lists of friends can also be opened from the friends list screen. In order to become friends with another account, the user must first send a friend request and have their request accepted by the other user. This is to ensure the privacy of the other individual.

Another option for the user is being able to look for other users and being able to look at other users' profiles. From this search function, the user can then choose to send a friend request to another account. The user can also have the search window list everyone by merely keeping the search bar blank and just hitting enter. The last option the user can do is being able to log out of the client by just pressing x on the menu.

In regards to the connection our application uses, it only creates temporary connections with the server when information needs to be sent to and from the server. This allows multiple users to connect to the server at once. In case the server gets overloaded, it has the ability to multi-thread if necessary.

Regarding the server, it calls a management class that handles storing all the data sent from the client to the server and also handles account retrieval. There is also a separate account class that deals with distributing information stored in users' profiles. Since all information that is inputting into the program is meant to be stored

whether the server is currently running or not, all the data is written to a file via printwriter after it's inputted into the application. This text file, called allUsers, is then called upon and updated while the server is running by the management class.