Katherine Gallaher 861100447

Kenneth Mayorga 860989982

Group 38

CS166 Project Report

To Start The Program:

In order to start the program you must have your Postgres instance started correctly with PGPORT=6400 and DB_NAME=projectdb. In order to load the data into the database enter the sql/src/load_data.sql file and change the paths to work for your computer and data. To load tables run the create_db.sh script in sql/scripts/create_db.sh. After you have loaded the database you can run the program by running the script compile.sh, no arguments needed, in java/scripts/compile.sh

High Level Description:

The following application allows the user: to create an account, change their passwords, update their profiles, send messages, view their emails, search for people in the network, send friend requests, reject friend requests, and view their friends' profiles. The application works as follows: first it asks the new user to create an account or to log in with a user name, email, and password. After successfully logging in, the user has the following choices to choose from a menu: go to their friends list, update their profile, display their profile, search for people, view or edit their messages and connection requests.

- If the user chose the option to update their profile, then the program allows them to include any of the following information: work experience, education details, date of birth, email, and their name.
- If the user chose the option of searching for people by name, then the program asks the user to enter the user id of the person they would like to search for. If the person exists, then the user can view that person's profile.
- If the user chose the option to send a friend request, then the program allows the user to send the request to anyone within the network if the user has less than 5 connection requests sent, otherwise the program only allows to send the request to people who are "3

levels deep". Within this option, the user can also manage their connections either by accepting or rejecting new friend requests.

- If the user chose the option to view messages, then the program allows them to view a list of their messages, and whether it is read or unread, and allows them to view contents and delete the message.

Problems Encountered:

A problem that was encountered was implementing the levels of a connection request. In order to do this you had to save all of the users that you were friends with, into a list, and then save all of the users that they were friends with. And then do this one final time for those friends. This involved using three for loops in the implementation. Another difficulty was deciding how to implement the messages. We had to decide whether to always show the message content or have the user decide to open the messages. We decided on allowing the user to open the message up to avoid clutter.

Design Decisions:

We chose to implement this system such that it was a series of lists in which users could select their option by number. We chose to do it this way because it is very easy for the user to understand how to use. Once the user is logged in they are given options of what they would like to do, again in a list. Another design decision we chose was whenever you visited someone's profile that was not your own you would have options to 1. send connection and 2. send message. Even if you already have a connection with the user whose profile you are visiting the option is still there. We decided to do it this way because it would be less confusing for the user if the interface did not change depending on whose profile you were looking at. We also decided that in order to send a message or a connection request you must be on a user's profile, you cannot send a message/connection unless you are on a profile. When you search for people on the network you are only able to search by user id and if that user exists you are immediately taken to their profile. This was done because we did not want to add too many extra submenus so we simply went directly to their profile.

Another design implementation that we decided on was the messages. We decided to implement them in such a way that you are able to view a list of all your messages, whether they are read or

unread, and then make another selection to read the contents. You cannot delete the message from this menu there is another option select for that. We did not want to confuse the user with too many buttons that is why we chose to implement it in that manner. We also decided to do connection requests in a similar manner as the messages, we show a list of all connection requests and the user can choose one and then choose to accept or deny the request.

Assumptions:

- When you search for users we have assumed that you are searching by user id instead of by the user's name, since a name is not required on sign up but a user id is.
- If a new user has no connections they are able to send up to 5 connection requests without the three levels of connection applying. However, once those 5 requests are sent the three level rule applies, whether any of those requests are accepted or not. You are always able to view the Send Connection option on a profile, however if it is selected and they are not within three levels you are given a message that you are not allowed to send a connection request.
- A user cannot delete or edit work or education experience after it is inputted, however if you have entered that menu unexpectedly you are able to exit.

Contributions:

For this project both partners worked together a lot of the time.

Specifically, Kenneth worked on the updating the profile, displaying the profile and searching for people and Katherine worked on the messages and connections. However, that does not mean that each person worked on only those parts, we worked together and helped each other in the implementation of all functions.