Project Phase 3
Jacob Jimenez
ID: 860-982-401
jjime022@ucr.edu
Katharina Kaesmacher
ID 613758
K.kaesmacher@gmx.de
June 8, 2015

CS 166 Database Management Systems
Instructor: V. Tsotras
Section: 021
TA: Karishma Dash
Group ID: 23

In completing this we consulted:

- The SQL Slides
- http://docs.oracle.com/javase/7/docs/api/java/util/Set.html (to understand how sets work in Java.)
- http://docs.oracle.com/javase/7/docs/api/java/util/Queue.html (to understand how queues work in Java.)
- http://en.wikipedia.org/wiki/Breadth-first_search (to better understand bread-first search.)

All the important code is original.

Contributions:

We worked on the entire project together so mentioning individual contributions is difficult. If one of us a wrote a function the other would correct mistakes and make suggestions for improvements. For example, Katie wrote the functions to Update Work/Educational Details and I fixed bugs such as the details not updating correctly. The workload was split 50/50 amongst us.

Design Decisions:

Updating the user's profile

a. Change Password
    - To update the password we require that the user enter their current password. Also, the user must enter the desired twice and if they don't match the password will not change. This was done to ensure the user's new password doesn't have a typo.
b. Update Name
    - The user is required to input the desired change in once. If the user makes a typo they can easily try again.
c. Update Email
    - Same implementation as Update Name.
d. Update Date of Birth
    - Same implementation as Update Name.
e. Update Work Experience
    - We created a sub menu for a user to update their Work Experience. We did this so the user can determine if they'd like to add a new details, or update their current details.
f. Update Educational Details
    - Similar implementation to Work Experience.

Messages:

- The menu messages allows the user to choose:
    a. Write a new message
        - To write a new message the user must input the userid. After the user will input their message and it will be sent.
    b. Read new messages
        - Will display any unread messages for the user.
    c. Read Received messages
        - Will display user's read messages.
    d. Read Sent messages
        - Will display the user's sent message. This includes messages that have been unread. The messages will displayed as a list. A prompt will ask if the user would like to delete a message. If the user inputs, Y, it will then ask to input the number of the message

they want to delete. If it's a valid number the message will be deleted.

Search People:

- A prompt appears asking for the user to input a name. It will return with a list of userid and names. Next it will ask for the user to select a profile by selecting the number associated with it. Then it will display the profile chosen. If the number entered is invalid the first profile will be displayed.
- When viewing another user's profile three options will appear:
    - Profile - If the current profile has friends this option will appear and allow the user to view another selected profile. If there are no friends displayed this option does not appear.
    - Message - Allows a message to be sent to the account being viewed.
    - Connection - Sends a connection request to the account. The connection request will only be sent if the user has five or less connections, or the account is no more than three levels away. Otherwise the connection request will fail and an output of "Not allowed to send request" is displayed.

Requests:

- The requests menu displays:
    a. Send Request
        - A prompt appears asking the user to input the userid. If the user has five connections or less the request will be sent. Otherwise, the request will only be sent if the account is at most three levels away from the user. If neither of the requirements are met the request will fail.
    b. Read Request
        - If the user has pending requests they will be displayed. The user then can choose to respond to a request by inputting the userid to respond to.

View Profile:

- This will display the user's profile showing their name, Education, Work Experience, and their friend list.

Challenges Faced:

- We were unable to load the Excel file into the database. We had to convert it into multiple CSV files, remove the special characters, and then we were able to load the data successfully.
- Figuring out how to implement the requirement that a connection request cannot be sent unless the user is a friend of a friend of a friend was difficult. We decided to use the breadth-first search algorithm. To do this we made a set named explored, and use two queues to determine which friend list to expand and search. We probably could have only used one queue, but due to time constraints we were unable figure out how.

Extra Credit:

- Good user interface. Our interface is easy to navigate and handles exceptions when a user inputs unexpected input.
- Indexes: We created an index on the USR table on name. This is a good index to have because when a user uses our search people function the database can quickly return the results.
- Our function that determines if a user is at most three levels away uses a set and queue to improve the performance of the search.

Assumption:

- To Update the date of birth function we assumed that the user wasn't born before 1901 and not born in 2016 or later.