CSE201: Advanced Programming
Lab Assignment 3

General Instructions:

- The lab is to be attempted in groups of 2.
- The TAs are there to help you out only with doubts in understanding the questions and probably some syntax related issue.
- Do not make use of internet while attempting the lab assignment.
- Do not indulge in any form of plagiarism as a strict penalty would be imposed.
- No deadline extensions would be made.

Submission Instructions:

- Make sure that you add the @author tag with your full name and roll number to every java file.
- Submit only the .java files. Keep the java files in a single folder and not in nested folders.
- Zip the files and submit on Backpack. Zip only the **.java** files and not the entire folder.
- Submissions not done in the described format shall not be evaluated.

Task:

Your task is to design a basic social network **(MyNetwork)**. The network consists of various Persons as represented by a Person class. The information about the person and his connections will be stored in a text file. Your program should have following functionalities:

1. User
    a. Register and login
    b. logout
2. Searching for friends by username
3. Adding friends from the network
4. Accepting and Rejecting friend requests
5. Listing friends associated with a Person
6. Listing mutual friends between two Person objects
7. Getting the Person's info (Refer to specific requirements)
8. Updating his/her status
9. Find the shortest route to a person in the network and display the route. **(Bonus)**

Specific Requirements:

- Note that, all the features except 1(a) require the Person to be logged in else you should prompt him/her to login and then it should proceed with the requested method.
- You are not required to design any GUI. Use the console to take username and password inputs while logging in.
- Use arraylists or simple arrays to store usernames instead of a hashable.
- Check for if the new user inputs the username which is already in the database.

- If a person is not your friend, you should be able to see just his/her mutual friends **(+shortest route (bonus))**.
- If a person is your friend, display his/her status,display name, friends, mutual friends.
- Username is unique, therefore if a person while signing up enters a username which is already in the database file, throw a user defined exception.
- If a person is not in the database when you search for it, the program should throw a user defined exception.
- All the changes of adding friends, accepting/rejecting friend requests, updating status, and other required database changes should be reflected in the database file.

- A sample user database in the form of comma (,) separated values is given to you.
- Your program should first read the database and build up the network.
- Then, it should prompt the user to login or register (sign up).
- The sample input file is in the following format:

<username>,<password>,<display_name>,<no_of_friends>,<user_name1>,<user_name2>,....,<no_of_pending_requests>,<user_name1>,<user_name2>,....,<Status>

- Here, pending requests are the friend requests sent by other users to you.
- Use character 'b' for Back.

Blue Marked: are inputs
Red Marked: are my comments

Sample of how the program should proceed for the scenario provided in the main method:

Reading database file…
Network is ready.

1. Sign up
2. Login

1

Enter username: bob_123
Enter display name: bob
Enter password: bob123

Registration is successful. User bob_123 created.

1. Sign up
2. Login

2

Please enter your username: mac_123
Please enter your password: mac123

mac logged in now.
Having fun in LA :D

1. List Friends
2. Search
3. Update status
4. Pending request
5. logout

1
Your friends are: cristina_123, samantha_123

1. List Friends
2. Search
3. Update status
4. Pending request
5. logout

2
Enter name: andrew_123
andrew_123 is not a friend.
Mutual Friends: No mutual friends
Shortest Route: mac_123 -> samantha_123 -> alice_123 -> andrew_123
    1. Send Request
    b. Back
1
Request sent.

andrew_123 is not a friend.
Mutual Friends: No mutual friends
Shortest Route: mac_123 -> samantha_123 -> alice_123 -> andrew_123
Request Pending.
    1. Cancel Request
    b. Back

b

1. List Friends
2. Search
3. Update status

4. Pending request
5. logout

2
Enter name: samantha_123

You and samantha_123 are friends

Display Name: Samantha
Status: house party rocks!!!
Friends: albert_123, drake_123, alice_123
Mutual friends: No mutual friends
b. Back

b

1. List Friends
2. Search
3. Update Status
4. Pending request
5. logout

3
Enter Status: Shun not the struggle.

Status updated!!

1. List Friends
2. Search
3. Update Status
4. Pending request
5. logout

4
1. lisa_123
2. alice_123
b. Back

2

alice
1. Accept
2. Reject

1
You are now friends with alice_123.

1. lisa_123
b. Back

1
lisa_123
1. Accept
2. Reject

2
Friend request by lisa_123 deleted.

No new friend requests.
b. Back

b

1. List Friends
2. Search
3. Update status
4. Pending request
5. Logout

2
Enter name: laila_123

User laila_123 not found. (This is the mandatory exception you need to catch and throw)

1. List Friends
2. Search
3. Update Status
4. Pending Request
5. Logout

5

User mac_123 logged out successfully

1. Sign up
2. Login