

Assignment 2 -- User database  
CSE231/506, Fall 2014  
Due: 11:55PM October 24, 2014

In this project, you will create a user database application. The application stores a list of users and a list of groups. A group has a list of users and a user is a member of one or multiple group. Each user must identify a major group it belongs to.

The user database application should be implemented with three classes: 1) UserDatabase class that uses ArrayList (or ArrayList2) implementation of ListInterface (Chapter 06) for the list of users and the list of groups; 2) User class that implements Comparable interface and an attribute named "id" of String type will be used as key to sort users; and 3) Group class that implements Comparable interface and an attributed named "name" of String type will be used as key to sort groups. The three classes should implements Serializable interface to allow easily read/write to/from files. To support serialization, I made minor modification to the ArrayList and other class from the class, mainly to make them implement Serializable interface. So please use the provided class from the repo (see below).

I have put the initial set of source files in bitbucket repo (<https://bitbucket.org/YonghongYan/cse231project/>) where you can clone the source folder from there using git utilities. For those who do not want to bother to learn using git, there is a download button on the webpage that you can click to download the zipped source tree. The readme.txt file in the root folder of the source tree tells you how to compile and launch the main function of the UserDatabase class. You can import the source tree to the Netbeans IDE. The three classes are in src/secs/cse231/project/ user package folder of the source tree.

Your task is to implement the methods that are included in the three classes. You are also allowed to add more methods or attributes to the classes to help your implementation or to add new features to this application. The methods include: 1). Basic member methods for manipulating objects of each of the three classes. I already provided the method signatures of most of these methods. Again, you are allowed to add more methods or attributes. (40 points)

For User class, you need to implement compareTo, equals, and toString methods (total 10 points); for Group class, you need to implement compareTo, equals, toString, containUser(String userID), and removeUser(String userID), and the Group constructor (total 10 points). For UserDatabase class, you need to finish the implementation of UserDatabase() constructor and removeUser(User u) method, implement the containGroup(String name), isGroupMember(String userID, String groupName), isGroupMember(User user, Group group), Group getGroup(String name), isGroupMember(String userID, String groupName), getMembers(Group g), getGroups(String userID) (for total 20 points).

2). Implement the main method that will provide a user interface (command line or GUI) for interacting with this database and perform operations (e.g. add/remove user, add/remove groups, query user/group info and memberships, etc). (30 points)

3). Implement the two functions for serializing and de-serializing the database from and to the file. (30 points), i.e. `readFromFile(String fileName)` and `writeToFile(String fileName)` (15 points each). The `SerSongsApp.java`, `SerSongList.java`, and `SerSong.java` classes of Chapter 06 provided very similar examples of using `ObjectInputStream` and `ObjectOutputStream` to read/write objects (even a list) from/to a file. Please refer to or using their codes to start the implementation of these two functions.

4). Authentication and authorization control: (40 points)

Authentication: A user can access the database only if he/she provides the correct user id and passwd, i.e. the implementation and use of login method in the project.

When the program starts, the main method should call login method to ask for a userID and password pair and compare them with those users already in the database. If they matches, the program continues for normal operations, otherwise, keep asking for userID/password pair or terminate (after giving a limited amount of trying). To bootstrap your application, you can turn off this feature so anybody can add users and groups, and then turn on this features after you have user in the database (you will need the serialization/deserialization feature functional to make this working).

Authorization: A user can only query the information of other users from the groups he/she also belongs to. You may also need to add a “root” user that has the unrestricted privilege to perform the database operations.

Your submission should include the following:

1. A document with the following contents
  - a. List all the validations you have done for example: your user name is validated and does not accept space or empty string as a value, adding a user that has the same user id as one already in this list is not allowed, the same rule for adding duplicated groups to the list, etc
  - b. List all the methods you have implemented and added to each file along with the file name
  - c. Include Output screenshots showing the following:
    - i. Disabling authorization and adding groups and users
    - ii. Enabling authorization
    - iii. Adding user, adding groups, removing user and removing groups
    - iv. Also include screen shots to show that your program validates all the input values you have mentioned
    - v. Querying for
      1. user and groups;
      2. users only from a specific group;
      3. all groups;
      4. all users;

5. all groups for a particular user
2. Source Code
  - a. Include a brief description or comments to the code you added describing the purpose and validations. It is highly recommended you submit your source folders and IDE configuration file together. Netbeans the preferred IDE you should use.

Your submission should be zipped in a single file named Assignment\_3\_<LastName><FirstName>.zip and submitted to Moodle.