Project: Advanced Social Network Management System

Overview

The **Advanced Social Network Management System** is a program designed to simulate a social networking platform. It allows users to create profiles, add multiple friends, create posts, and share posts with specific friends or all friends. The system uses **linked lists** to manage users, friendships, and posts efficiently. It also provides reporting features to track user activity, post engagement, and social connections. See figure 1.

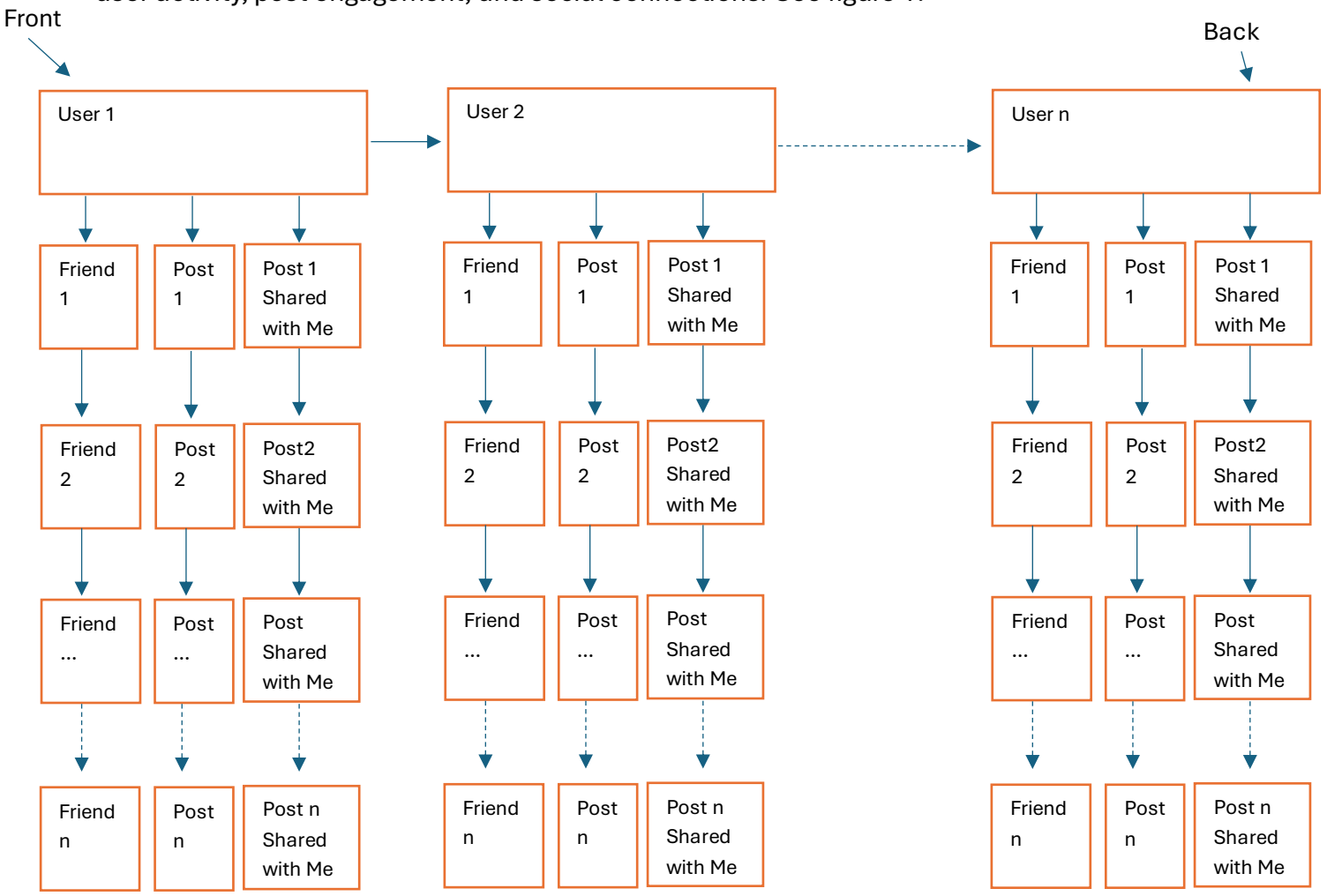

Figure1: User Profiles: Abstract Overview

As a student in the COMP242 course, you are required to implement this system while considering the following key features.

Key Features

1. File Management

- The program reads three input files:
    1. **users.txt**: Contains user data (`user ID, name, age`).
    2. **friendships.txt**: Contains friendship data (`user ID followed by a list of their friends`).
    3. **posts.txt**: Contains post data (`post ID, creator ID, content, creation date, and shared user IDs`).

Example of File 1 (users.txt): {This is only example. Create a large file}

```
User ID,Name,Age
1,Ahmed,25
2,Fatima,30
3,Khaled,22
4,Layla,28
```

Example of File 2 (friendships.txt): {This is only example. Create a large file}

```
User ID,Friends
1,2,3,4
2,1,3
3,1,2
4,1
```

This means:

- User 1 (Ahmed) is friends with users 2, 3, and 4.
- User 2 (Fatima) is friends with users 1 and 3.
- User 3 (Khaled) is friends with users 1 and 2.
- User 4 (Layla) is friends with user 1.

Example of File 3 (posts.txt): {This is only example. Create a large file}

```
Post ID,Creator ID,Content,Creation Date,Shared With
1,1,Hello World,21.3.2025,2,3,4
2,2,Ana saeeda alyawm!,15.3.2025,1
3,3,Salam Alaikum,22.3.2025,1,2
```

- The program includes a **file chooser** to allow users to select and read these input files.
- It handles file read errors and data format inconsistencies with **robust exception handling**.

2. User Data Management

- Each user is represented as a node in a linked list, with the following attributes:
  - **User ID**: Unique identifier for the user.
  - **Name**: Full name of the user.
  - **Age**: Age of the user.
  - **Friends**: A linked list of friends (other users).
  - **Posts Created**: A linked list of posts created by the user.
  - **Posts Shared with them**: A linked list of posts shared with the user by their friends.
- **Example**:
  - **Ahmed**:
    - Friends: Fatima, Khaled, Layla
    - Posts Created: Post ID 1 ("Hello World ")
    - Posts Shared with him: Post ID 2 ("Ana saeeda alyawm!"), Post ID 3 ("Salam Alaikum ")

- The system supports the following operations on user records:
    - **Insert**: Add new users to the system.
        - Example: Add a new user with ID 5, name Sara, and age 24.
    - **Delete**: Remove users from the system.
        - Example: Delete user ID 4 (Layla) after confirming the action.

          Please note that if a user is deleted from the system, all posts created by that user should be removed from the system entirely, including from the profiles of those with whom it was shared. Furthermore, the deleted user must also be removed from the friend lists of all other users they were connected with.
    - **Update**: Modify existing user information.
        - Example: Update user ID 1 (Ahmed) to change their age from 25 to 26.
    - **Search**: Efficiently locate users by ID or name.
        - Example: Search for user ID 2 (Fatima) and display their details.

3. Friendship Management

- The system allows users to:
    - **Add Friend**: Add a friend to a user's linked list of friends.
        - Example: User ID 5 (Sara) adds User ID 1 (Ahmed) as a friend.
    - **Remove Friend**: Remove a friend from a user's linked list of friends.
        - Example: User ID 1 (Ahmed) removes User ID 4 (Layla) as a friend.

4. Post Management

- The system allows users to:
    - **Create Posts**: Users can create posts and share them with specific friends or all friends.
        - Example: Ahmed creates a post with content " I'm enjoying the COMP242 course :) " and shares it with Fatima, Khaled, and Layla.

- **View Posts**: Users can view posts shared with them by their friends.
    - Example: Fatima views posts shared with her by Ahmed and Khaled.
- **Delete Posts**: Users can delete their own posts, but not others. For example, I can't delete a post shared with me from the entire system, but I can remove it from my view. If the owner deletes the post, it should also be removed from the accounts (profiles) of all friends it was shared with.
    - Example: Ahmed deletes his post with ID 1.

5. Reporting and Statistics

- The system generates the following reports:
    - **Posts Created by User**: Display all posts created by a specific user.
        - Example: Display all posts created by Ahmed.
    - **Posts Shared with User**: Display all posts shared with a specific user.
        - Example: Display all posts shared with Fatima.
    - **Most Active Users**: Display the top N users with the most posts (Active users who have published a post in the past three weeks or users who have more posts).
        - Example: Display the top 2 most active users.
    - **Engagement Metrics**: Track the number of posts each user has created and shared.
        - Example: Ahmed has created 1 post and has 2 posts shared with him.

6. User Interface

- The graphical user interface (GUI) includes:
    - A **table view** to display user, friendship, and post data clearly.
    - **Navigation buttons** (Next) for easy record browsing.
    - **Confirmation messages** for critical actions, such as deletions and updates.

- A **statistics section** to display reports (see the **Reporting and Statistics** above (point 5 / page 5)).
- **Different menus** (for example, Post menu, User menu, among other menus). You are autonomous. Do not ask the instructors about how many menus you need to add to the system or which menus should be included. Analyze the system and add what you think is appropriate
- **Use date/time picker to enter the date and not manually**.

You may be asked during the discussion to print the friends of a specific user, sorted in either ascending or descending order based on their username. Make this feature work. See Visualization of Linked Lists section below (page 8)

7. Data Management

- The program can:
  - Display reports on the user interface or save them to a file.
    - Example: Save the list of posts created by each user to posts_created.txt.
  - Save the updated user, friendship, and post lists back to the files.
    - Example: Save updated post data to updated_posts.txt.

Please add an option to save data as either **unsorted or sorted by username, in either ascending or descending** order (use a combo box to allow selection of ascending or descending).

Sample Output (Sorted by username/ ascending)

Posts Created Report (posts_created.txt)

```
Posts Created Report

--------------------

User: Ahmed

- Post ID: 1, Content: Hello World, 21.3.2025, Shared With: Fatima, Khaled, Layla

User: Fatima

- Post ID: 2, Content: Ana saeeda alyawm!, 15.3.2025, Shared With: Ahmed


User: Khaled

- Post ID: 3, Content: Salam Alaikum, 22.3.2025, Shared With: Ahmed, Fatima
```

Posts Shared With User Report (posts_shared.txt)

```
Posts Shared With User Report

----------------------------

User: Ahmed

- Post ID: 2, Content: Ana saeeda alyawm, 15.3.2025, Creator: Fatima

- Post ID: 3, Content: Salam Alaikum, 22.3.2025,Creator: Khaled


User: Fatima

- Post ID: 1, Content: Hello World, 21.3.2025, Creator: Ahmed

- Post ID: 3, Content: Salam Alaikum, 22.3.2025, Creator: Khaled


User: Khaled

- Post ID: 1, Content: Hello World, 21.3.2025, Creator: Ahmed

- Post ID: 2, Content: Ana saeeda alyawm!, 15.3.2025, Creator: Fatima
```

Visualization of Linked Lists (An example of how you can imagine the system)

Ahmed:

- Friends: [Fatima] -> [Khaled] -> [Layla] -> NULL
- Posts Created: [Post ID: 1, Content: Hello World, Date: 21.3.2025] -> NULL
- Posts Shared With Him: [Post ID: 2, Content: Ana saeeda alyawm!, Date: 15.3.2025] -> [Post ID: 3, Content: Salam Alaikum, Date: 22.3.2025] -> NULL

Fatima:

- Friends: [Ahmed] -> [Khaled] -> NULL
- Posts Created: [Post ID: 2, Content: Ana saeeda alyawm!, Date: 15.3.2025] -> NULL
- Posts Shared With Her: [Post ID: 1, Content: Hello World, Date: 21.3.2025] -> [Post ID: 3, Content: Salam Alaikum, Date: 22.3.2025] -> NULL

Khaled:

- Friends: [Ahmed] -> [Fatima] -> NULL
- Posts Created: [Post ID: 3, Content: Salam Alaikum, Date: 22.3.2025] -> NULL
- Posts Shared With Him: [Post ID: 1, Content: Hello World, Date: 21.3.2025] -> [Post ID: 2, Content: Ana saeeda alyawm!, Date: 15.3.2025] -> NULL

Please note the Followings:

I. Your application should have all functionalities working properly.

II. All the files mentioned in this project **are examples**. You need to generate your own file with at least 30 users and 7 posts/user.

III. There must be adequate documentation and comments in the code (e.g., functions, loops, etc.).

IV. Your code should follow coding conventions (e.g., spacing, indentation, etc.) and guidelines (**Remember COMP2311**).

V. This is an **individual Project**. Disciplinary action will be taken against those who **cheat**. Additionally, the use of **AI tools** for generating solutions or **copying from websites** is strictly prohibited. Students found in violation of these policies will face severe consequences. It is crucial to ensure that all work submitted is **your own** and adheres to the **guidelines provided for this project**.

VI. During the discussion**, don't say, "I forgot because I worked on the project one or two weeks ago"**. If you do, **you will receive a zero for the project**. A discussion means **you must be ready and prepared beforehand**. You will be asked about all the functionality, in addition to time and possibly space complexities.

VII. Please submit your Java files (java) and corresponding test text files (txt) via the ITC by Monday, 7/ 4/2025, at 11:00 PM. Late submissions will not be accepted under any circumstances.

All the Best :)