Machine Coding Question (FK Social Feed)

Problem Statement

Build a console social media application for flipkart that allows users to upload posts like pictures, blogs etc., interact with others' posts, and follow other users on the platform.

Functional Requirements

1. **User Registration**: A new user should be able to register with an unique id and username.
2. **Upload a Post**: A user should be able to upload a post of their choice (For simplicity, consider the post as a string value).
3. **Follow/Unfollow Users**: A user should be able to follow or unfollow any number of other users.
4. **Like/Dislike Post**: A user should be able to like or unlike any number of posts.
5. **Show Feed for a User**: A method to view posts uploaded by other users, sorted based on the following criteria(in order of priority):
    a. Posts from users that the current user follows will always appear at the top of the feed. Posts from non-followed users will come after.
    b. Within each group of followed and unfollowed, posts should be sorted by score(likes - dislikes)
    c. If two posts have the same score, they are sorted by posting time, with the more recent post appearing first

Example Usage

1. **Register User:**
   RegisterUser <user_id> <user_name>
    a. Input: RegisterUser 1 Akash // Register Akash
       Output: Akash Registered!!
    b. Input: RegisterUser 2 Hemant // Register Hemant
       Output: Hemant Registered!!

2. **Upload a Post:**
   UploadPost <user_id> <post>
    a. Input: UploadPost 1 "This is my first post. My name is xyz" // User with id 1 uploads a post
       Output: Upload Successful with post id: 001
    b. Input: UploadPost 1 "I work at flipkart as a SDE1" // User with id 1 uploads another post
       Output: Upload Successful with post id: 002

3. **Like/Dislike a Post:**
   InteractWithPost <interaction_type> <user_id> <post_id>
   a. Input: InteractWithPost LIKE 2 001 // User2 likes post with id 001
      Output: Post Liked!!
   b. Input: InteractWithPost DISLIKE 2 001 // User 2 dislikes post with id 001
      Output: Post Disliked!!

4. **Follow/Unfollow a User :**
   InteractWithUser <interaction_type> <user_id1> <user_id2>
   a. Input: InteractionWithUser FOLLOW 2 1 // User2 follows User1
      Output: Followed Akash!!
   b. Input: InteractionWithUser UNFOLLOW 2 1 // User2 unfollows User1
      Output: Unfollowed Akash!!

5. **Show feed for a User:**
   ShowFeed <user_id>
   a. Input: ShowFeed 1
      Output: UserName -
      # of Likes -
      # of Dislikes -
      Post -
      Post time -

```
UserName - Akash
# of Likes - 2
# of Dislikes - 1
Post - I work at flipkart as a SDE1
Post time - 2024-10-18 10:11 PM

UserName - Hemant
# of Likes - 1
# of Dislikes - 10
Post - I too worked at flipkart as a SDE1
Post time - 2024-10-18 10:101 PM
.
.
```

Bonus Functionalities (Optional)

1. **User Tagging in Posts**: Allow users to tag others and notify the tagged users.

   InteractWithPost <interaction_type> <user_id> <post_id> <comma_ids_to_tag>

Ex: InteractWithPost TAG 2 001 1,3 // User2 tags User1, User3 on the Post 001

2. **Display Relative Time:** Show time in a relative format such as "2m ago" or "1 hr ago"**.**

   Output: UserName -
   # of Likes -
   # of Dislikes -
   Post -
   Post time - 2m ago

3. **Comment on Post:** Allow users to comment on others' posts. Add list of comments as additional attribute for each post in showFeed functionality's output.
   InteractWithPost <interaction_type> <user_id> <post_id> <comment_string>

   Ex: InteractWithPost COMMENT 1 001 "Can you refer me??"

Guidelines

1. Input can be read from a file or STDIN or coded in a driver method. **[No Api and No UI]**
2. Output can be written to a file or STDOUT. **[No Api]**
3. Store all interim/output data **in-memory data structures**. The usage of databases is not allowed.
4. Limit internet usage to syntax lookup only.
5. Language should be **Java only**.
6. Save your code/project by your name and email it or upload on the google form link provided. Your program will be executed on another machine. So, explicitly specify dependencies, if any, in your email.
7. Duration for coding will be **120 minutes.**

Expectations

1. Create the sample data yourself. You can put it into a file, write unit tests case or within the main driver program itself.
2. **Code should be demo-able (very important)**, either by using a main driver program or test cases. The code should be **functionally correct and complete**. At the end of this interview round, an interviewer will provide multiple inputs to your program for which it is expected to work.
3. **Bonus functionalities** should be implemented once the core functionalities are complete.
4. **Avoid writing monolithic code**.
5. The Code should be **readable, modular, testable, extensible with proper naming conventions**. Use intuitive names for your variables, methods, and classes.

6. The Code should **handle edge cases properly and fail gracefully**. Add suitable exception handling with proper messaging wherever required.
7. It should be **easy to add/remove functionality** without rewriting the entire codebase i.e. extensible for new features.
8. The code should follow **object-oriented design** principles.
9. **Don’t use any database, store all the data in memory.**