**Progetto per il corso di Algoritmi Distribuiti**

Federico Bertoli Matricola: 157896

**Introduzione**

Per questo progetto ho deciso di implementare una semplice replica del social network Twitter, utilizzando Java RMI.

Immagine che contiene testo, schermata, software, Carattere

Descrizione generata automaticamenteLa struttura del progetto è composta dai componenti principali Server e Client, metodi di utilità aggiuntivi come il file di costanti, i modelli dei dati scambiati tra le componenti, il DAO (Data Access Object) e l’interfaccia dei metodi implementati dal server.

Le funzionalità sono le seguenti:

**Server**

* Registrazione al servizio (creazione handle come Twitter)
* Login utente
* Scrittura post
* Commento di un post
* Possibilità di aggiungere un like ai post
* Possibilità di seguire le persone (concetto di following) e dunque di essere seguiti (concetto di followers)

Figura 1 Struttura del progetto

* Recupero dei soli post delle persone che si seguono con e senza filtri temporali
* Recupero di tutti i post con e senza filtri temporali

**Client**

* Implementazione di una UI apposita da terminale per sfruttare le funzionalità offerte
* Cache per evitare chiamate di rete superflue
* Scambio di messaggi tra utenti

**Metodi implementati e descrizione**

Il server implementa la seguente interfaccia:

**Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente**

Ad ogni operazione effettuata sugli utenti e sui post, ove necessario, le informazioni vengono salvate su file per garantirne la persistenza al riavvio.

**registerUser**(String userHandle):

Permette la registrazione dell’utente. È presente un controllo sull’esistenza dello stesso handle per evitare duplicati.

**login**(String userHandle):

Permette l’accesso ad un utente già registrato. Se l’utente non è presente, viene invitato ad effettuare prima la registrazione.

**newPost**(String userHandle, String post):

Crea un nuovo post, aggiungendolo alla lista di quelli già presenti.

**deletePost**(String userHandle, String postUuid):

Permette l’eliminazione di un post; questa è possibile solamente se è l’utente che ha creato il post che ne richiede l’eliminazione.

**likePost**(String userHandle, String postUuid):

Aggiunge un like al contatore di ogni post, visibile nell’interfaccia utente.

**followUser**(String follower, String followed):

Permette di seguire un utente. In questo modo, dall’interfaccia è possibile recuperare solamente i post degli utenti che si seguono, come su Twitter. È presente il controllo per evitare che un utente possa seguire sé stesso.

**unFollowUser**(String follower, String followed):

Permette la rimozione di un utente dalla propria lista di utenti seguiti.

**getLatestPosts():**

Recupera la lista di tutti i post presenti fino a quel momento, senza distinzione di utenti

**getFollowedPosts**(String userHandle):

Recupera solamente i post degli utenti seguiti da quel particolare utente

**commentPost**(String userHandle, String postUuid, String comment):

Aggiunge un commento al post selezionato; questo sarà visualizzato al di sotto del post. È possibile aggiungere un numero potenzialmente infinito di commenti.

**getUsersList**(String userHandle):

Recupera la lista degli utenti e le informazioni dei loro client, per permettere la comunicazione diretta utente – utente.

**directMessage**(String userHandle, String message):