[CHAR] = {a, ... , z} ∪ {A, ..., Z} ∪ {0, ... , 9}

Id
X: ℕ

Timestamp
X: ℕ

User
ID: Id
Name: N ⊆ CHAR
Password: P ⊆ CHAR

|Name| > 0

[USER_ID] = Id / Id ∈ { User }

Tweet
ID: Id
Text: CHAR
AuthorId: Id / Id ∈ USER_ID
At: Timestamp

|Text| > 0
|Text| ≤ 280

UserFollows
From: Id / Id ∈ USER_ID
To: F ⊆ USER_ID

UserTimelineCache

stdOut == error | accepted

Register

Δ User
name?: N ⊆ CHAR
password?: P ⊆ CHAR
out!: stdOut

---

name? ∈ dom(user) ⇒ out! = error
User' = User ∪ {
     {ID: Id / Id.X = max(USER_ID) + 1, name, password }
  }
  ⇒ out! = accepted

NewTweet

Δ Tweet
authorId?: Id
authorPassword?: K ⊆ CHAR
text?: T ⊆ CHAR
out!: stdOut

{authorId, authorPassword} ∉ User ⇒ out! = error
newTweet = {ID: Id / Id.X = max(TWEET_ID) + 1, text, authorId?, At: now()}
Tweet' = Tweet ∪ { newTweet }
UserTimelineCache' = UserTimelineCache ∪ { newTweet }
| UserTimelineCache' | ≤ 5 ⇒ out! = accepted
UserTimelineCache' = UserTimelineCache - B / B =
          min(UserTimelineCache.At)
    ⇒ out! = accepted