**Twitter Design**

Requirements:

1. User Login / signup authentication.
2. Users should be able to see all the posts made by their followers.
3. Users should be able to post tweets.
4. Users should be able to like and comment on tweets.
5. Users should be able to search posts based on username and should be able to search users.
6. Users should be able to follow and unfollow other users.
7. Users can delete their tweets.
8. Users can delete their accounts.
9. Users can edit their profile and have profile pictures.

Constraints:

1. Only tweets of followers will be shown.
2. No recommendation system.
3. No chatting system.
4. No media uploads on tweets.

Capacity:

1. Users = 1 billion
2. Daily active users: 100 million users.
3. Considering each user\_metadata is of 1KB and max profile picture limit is 5MB.
4. Total data for users required is 5 PB.
5. Considering per day a single user makes 10 tweets. = 1000 million = 1 billion tweets/day.
6. Tweets Metadata space: Considering 1 tweet requires 100 B of data. = 100 GB/ day per replication.
7. Considering comments on each tweet is 1000 on average = Total 1 trillion comments/ day.

Components:

1. Users Database
2. Tweets Database
3. Comments Database
4. Load Balancer
5. CDN
6. Web Servers

**High level Design**



APi’s

1. Login / SignUp and authentication API.
2. Edit profile details API.
3. Creating tweets API:
   1. If a Celebrity user posts a tweet.
   2. If a normal user posts a tweet.
4. API to get all followers of a user by given user id.
5. API to get all following lists of a user by the user id.
6. API to fetch tweets:
   1. Tweets by celebrities that a user follows
   2. Tweets by users .
7. API to fetch comments based on tweet id.
8. APi to delete an account.
9. API to delete a tweet
10. API to delete a comment.

Database Design:

1. Users: *(user\_id, username, password, age, email, location)*
2. Followers: *(user\_id, follower\_id)*
3. Following: *(user\_id, following\_id)*
4. Tweets: *(user\_id, tweet\_id, content)*
5. Comments: *(comment\_id, tweet\_id, comment)*