Department of Computer Science and Engineering

The Chinese University of Hong Kong

CSCI3100

DFD Specification Document

LITTLE TWITTER – GROUP F1

1155157657 CHAN SIU CHUNG

1155109984 CHOW CHUN MAN

1155175870 CHENG TSZ KIN

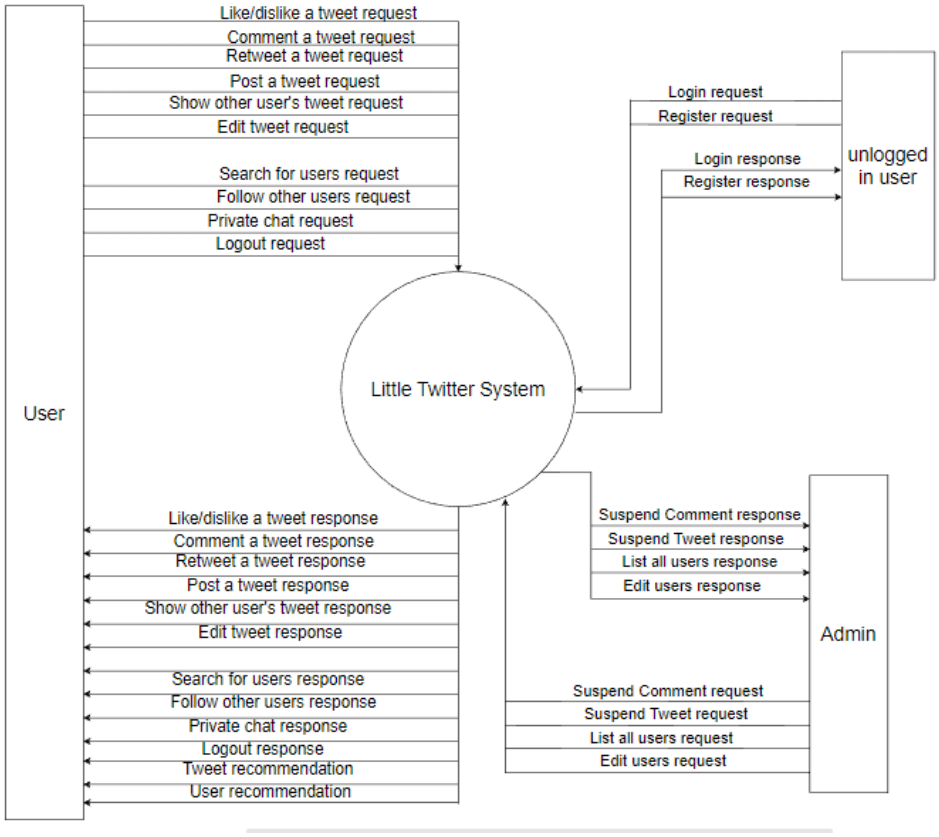
1155158936 HO CHUN WAI

1155172012 HUI ROCKY

2023-02-25

Contents

1. High-Level Context Diagram

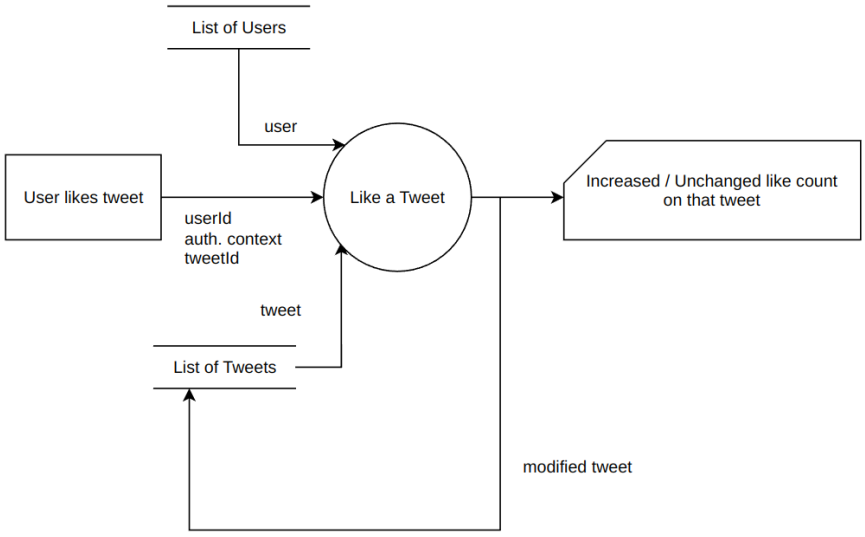


Description:

This is the high-level context diagram on our Little Twitter website. There are three types of users in Little Twitter System. They are User, unlogged-in user and Admin. Each of them has corresponding requests that can be sent to the system. The functionalities will be discussed in the section of feature diagrams.

2. Feature Diagrams

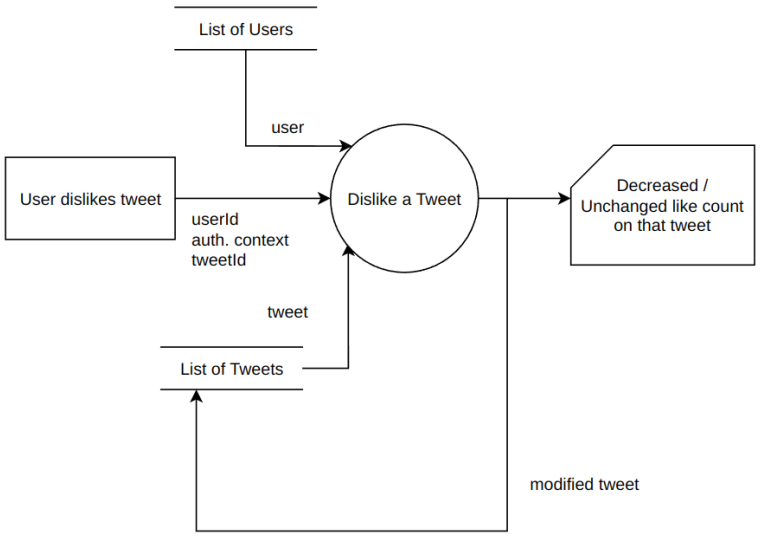
2.1. Like/dislike a tweet



Description:

When an authenticated user likes a tweet, the tweet id and user id are sent to the server. By the ids the function can fetch the required tweet and user. We increase the like count if the user has not liked the tweet yet. After the modification the tweet is stored back in the database.

The user interface reflects the changes by updating the like count.



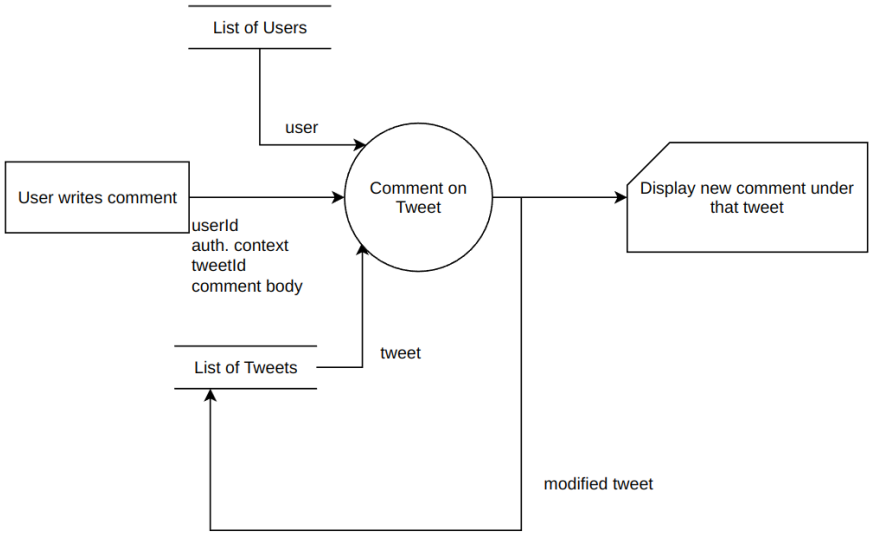
Description:

When an authenticated user dislikes a tweet, the tweet id and user id are sent to the server. By the ids the function can fetch the required tweet and user. We decrease

the like count if the user has not liked the tweet yet. After the modification the tweet is stored back in the database.

The user interface reflects the changes by updating the like count.

2.2. Comment a tweet



Description:

When an authenticated user writes the comment and sends it, the user id, tweet id and comment body will be sent as arguments. The function pulls the user and the tweet that will contain the comment and modifies the tweet by pushing the comment body to the list of comments. Finally, it stores the modified tweet back to the database.

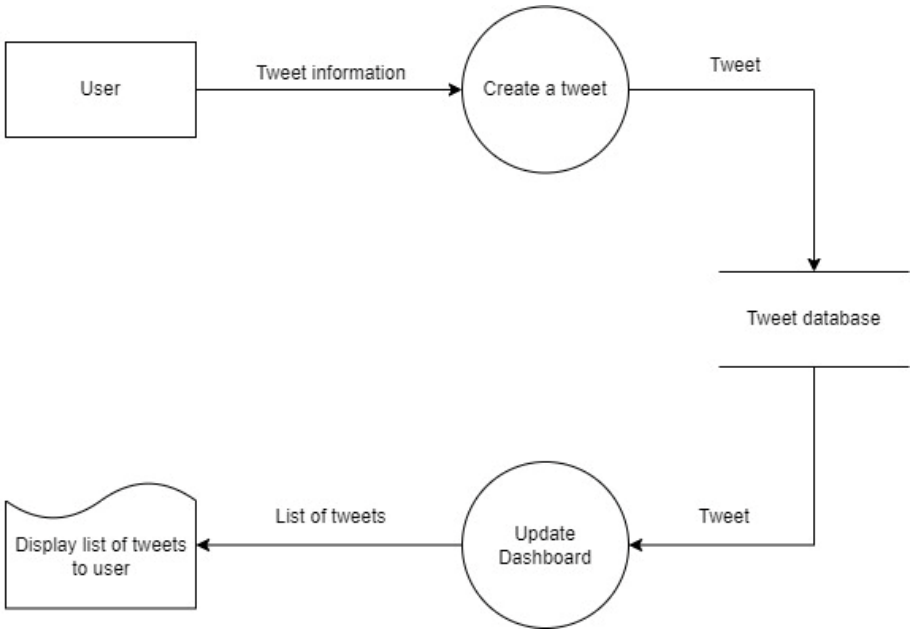
The user interface reflects the changes by displaying a new comment under the existing comments.

2.3. Post a tweet

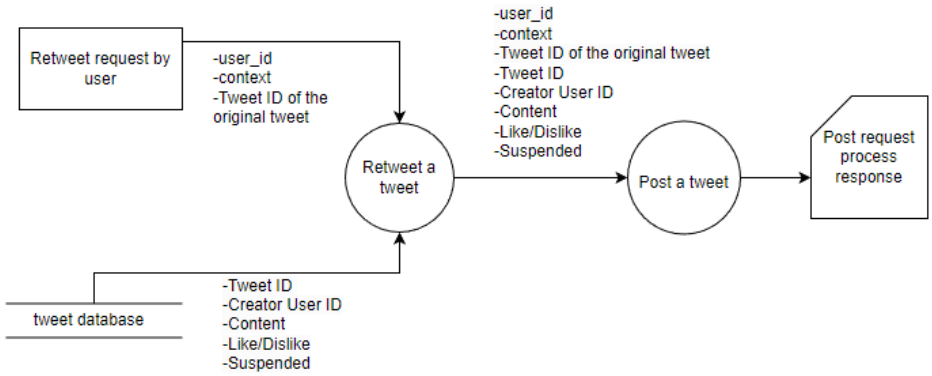
Description:

Users can post a tweet by inputting tweet information. The server will store Tweet object to the database and updates user's dashboard

DFD:



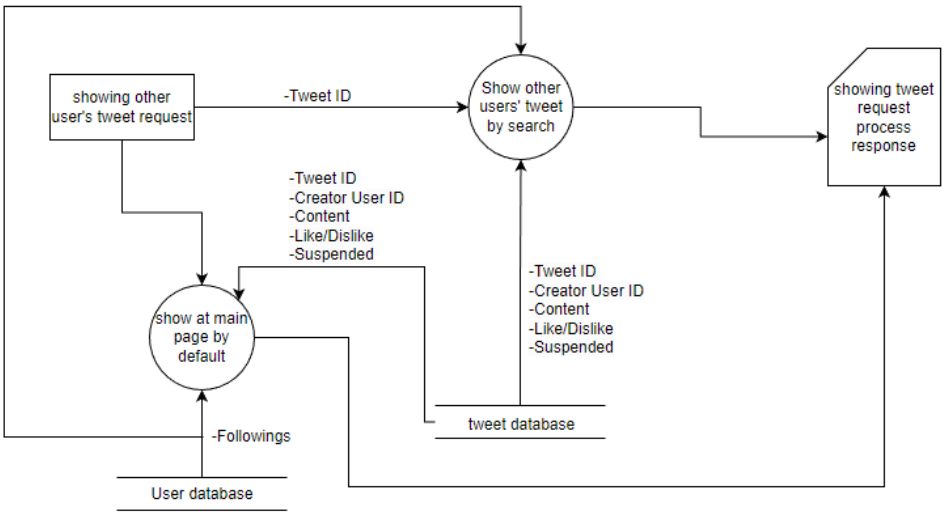
2.4. Retweet a tweet



Description:

Often people will share the tweets created by other users. We implement retweet feature by using the “post a tweet feature” mentioned before.

2.5. Show other users' tweet



Description:

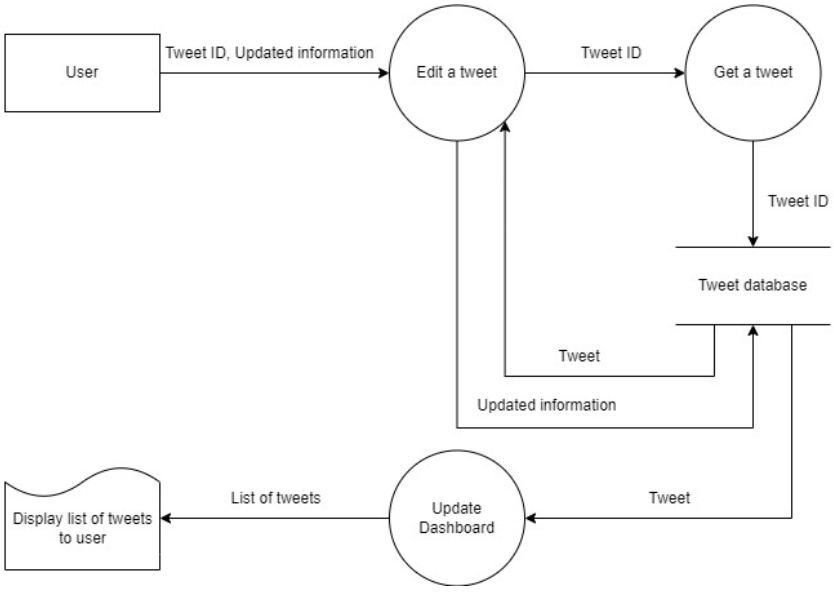
In our Little Twitter system, we should show users the tweets of whom they are following on the main page if they are not suspended by the system. So, we would extract the necessary information from different databases and show it to users. It is the same when it comes to searching a tweet with tweet id.

2.6. Edit Tweet

Description:

User can edit an existed tweet by click edit button, then the server will receive the Tweet ID, and sent get tweet request to database. After that, the database will response the tweet object. So that user can sent updated information to the server and store to database. Finally, the dash board will be updated and display the latest information to user.

DFD:

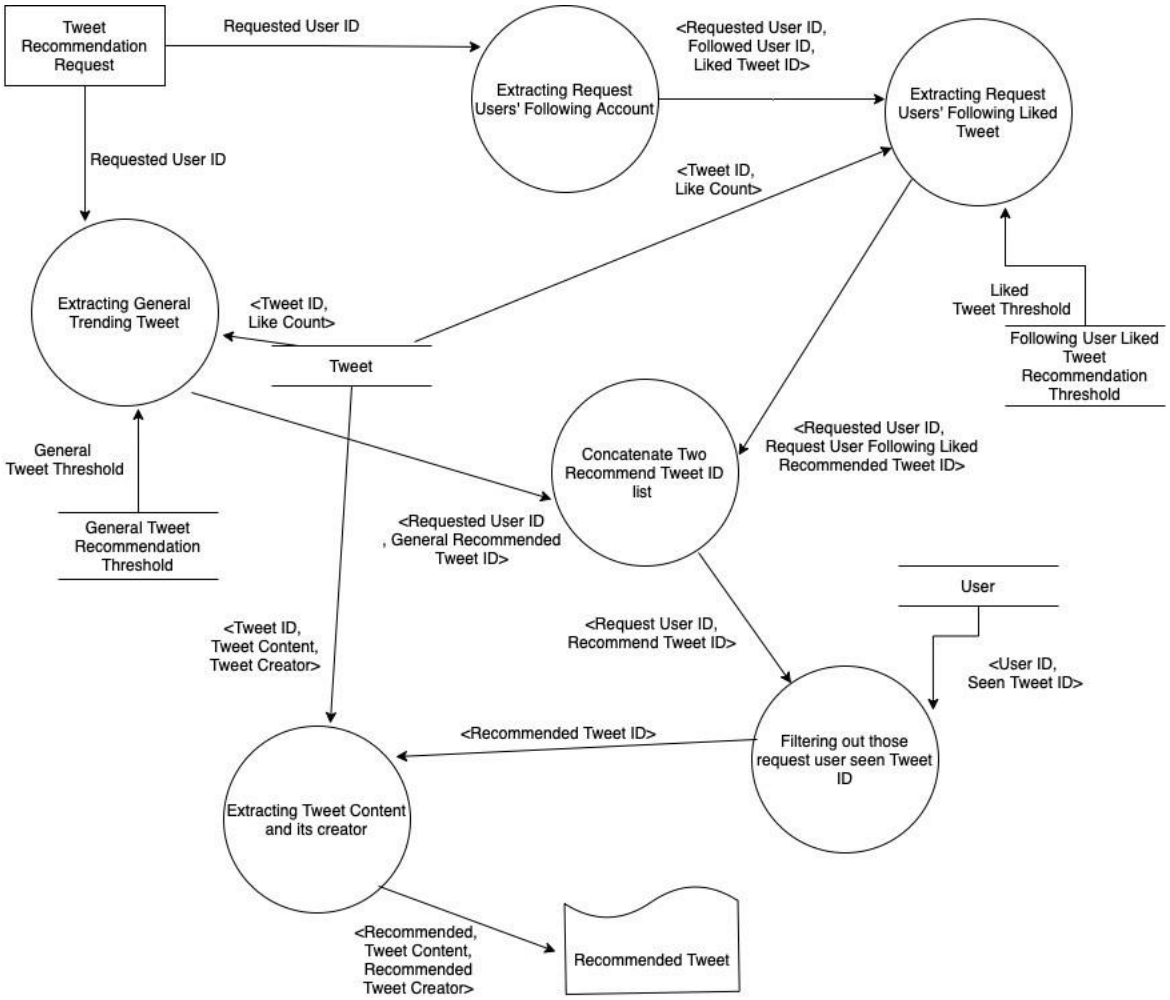


2.7. (Advanced) Tweet Recommendation

Description:

In our Twitter software system, we would like to post the tweet overview on the users' homepage. Therefore, users could quickly get an update on the recent status or situation of other users that he/she followed. Besides, we would also like implement tweet recommendation for the tweet overview. This would allow user to get to know other users that he/she does not follow.
There are two schemes for recommending a tweet to the user.
First, we would set up a threshold for the Like Counts. If the tweet has the Like Counts over the threshold, we would recommend the tweet to all users that does not read it before no matter whether tweet creator and tweet receiver follow each other. We called those recommended tweets as general recommendations.
Secondly, we would recommend the tweet based on the behavior of the account that followed by the users. A threshold would also be set up so that if the tweet receives the Like Count from the following users over the threshold and the user does not see it before, we would recommend the tweet to them. We call those recommended tweets as user-specific recommendations
Note that user-specific recommendations would be ranked higher than general recommendations. Within these types of recommendations, the recommended tweet would be ranked by the like count

DFD:

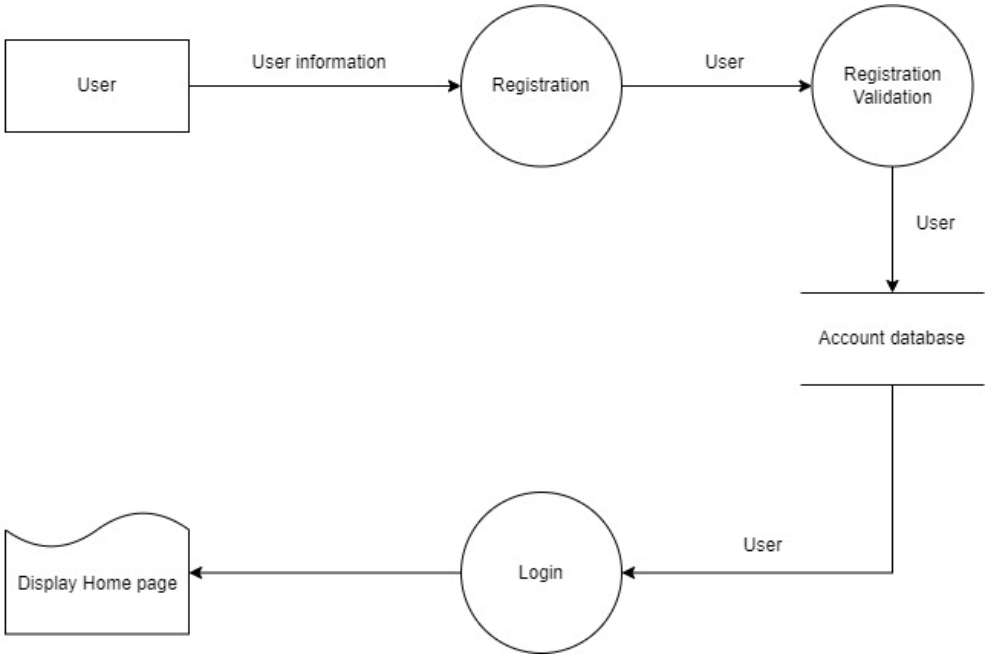


2.8. Register

Description:

A guest user can register for an account by input required information. The server will validate and create an account to database. Then, it will do login process and the user will be navigated to home page.

DFD:

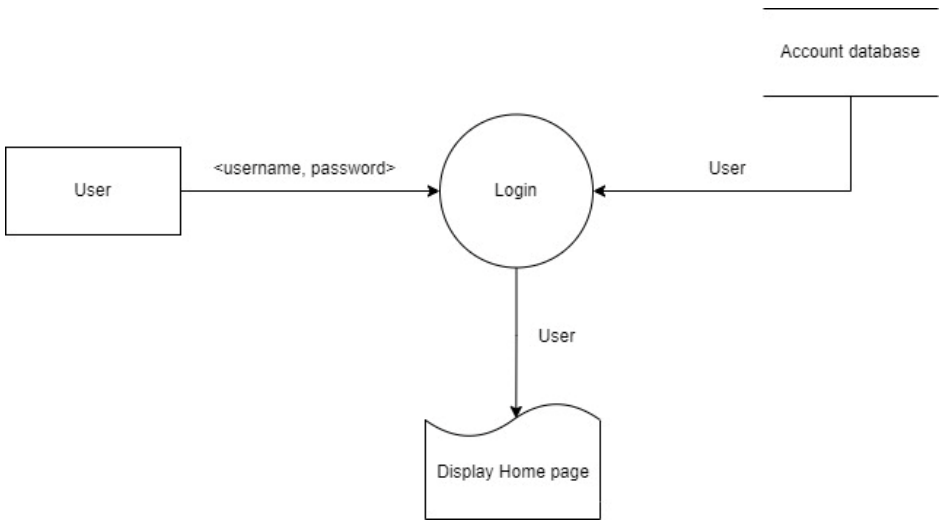


2.9. Login/Logout

Description:

A user can do login process by inputting username and password at login page. Then, account database will return the required user object and navigate the user to his/her home page.

DFD:

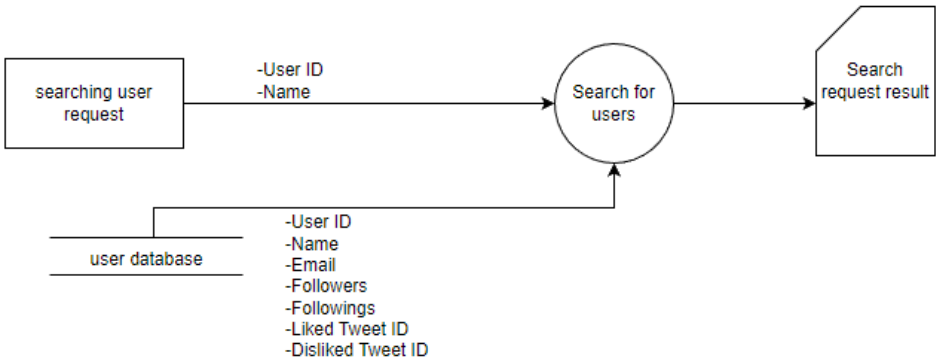


2.10. Search for users

Description:

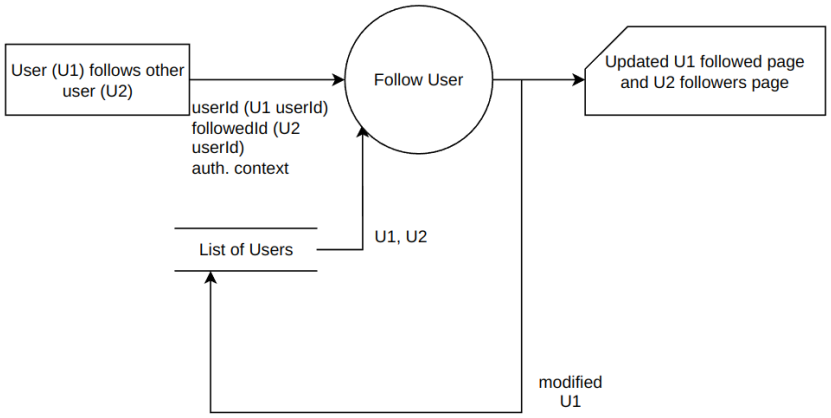
A user can search for another use based on the unique user ID or name. If the user has already followed the searched user, then he/she can view more information.

DFD:



.

2.11. Follow other users



Description:

When the user clicks the follow button, the authenticated user id and followed user id is sent to the function. The function fetches both users from the database. Since we are using a NoSQL database, the only user we have to modify is the authenticated user. The function simply pushes the followed user id into the followed array, then it stores the user back to the database.

The user interface will have two changes. The authenticated user's followed page will have the followed user listed, and the followed user's followers page will have the authenticated user listed.

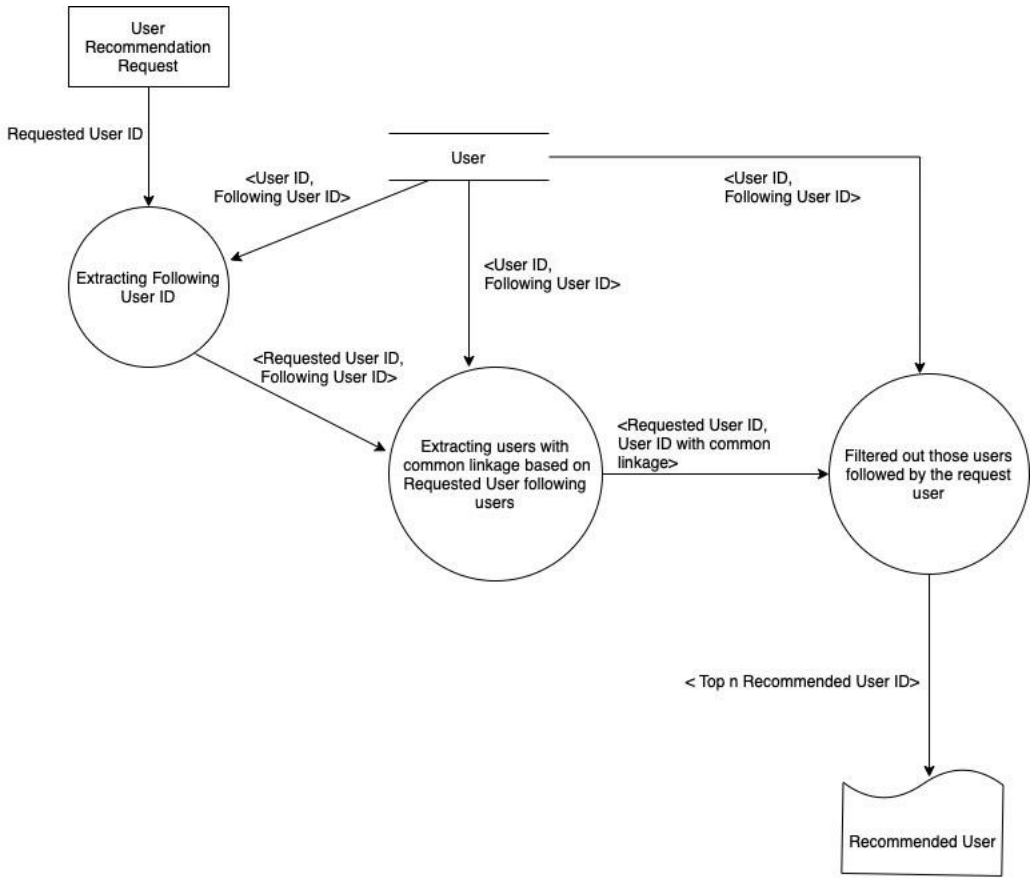
2.12. User Recommendation

Description:

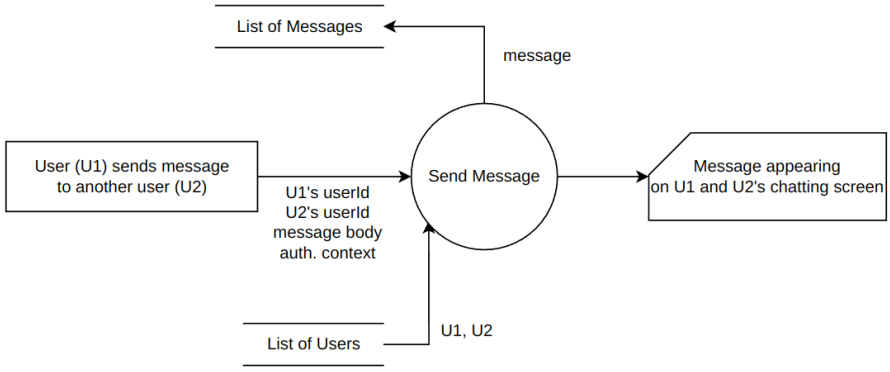
In our Twitter software system, we would like to allow the users to follow each other which can enhance the linkage between users and increase their sense of belongings to our software. Besides, we may also extract more information about the users via his/her linkages to enhance the user experience. Therefore, we would implement users recommendations which would provide recommendations on users that he/she would like to connect with. This would facilitate our data mining

We would provide users recommendations based on the primary user's existing following accounts. We would look at the common following users of those existing following accounts. For those users with the highest common linkage and the primary user does not follow that users, we would recommend that users to the primary user

DFD:



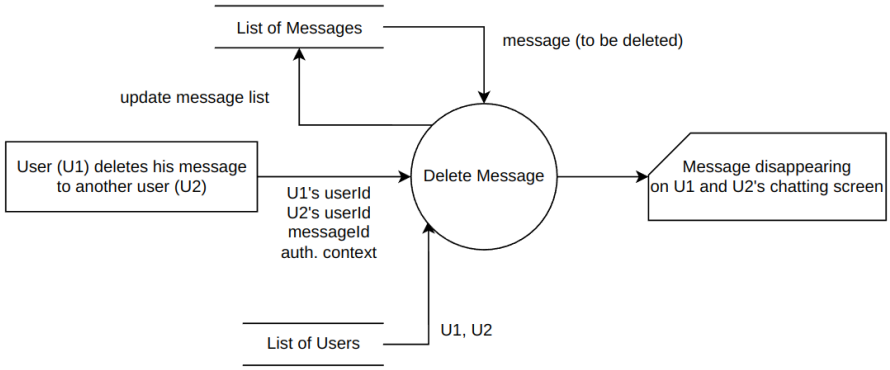
2.13. Private Chat



Description:

When the authenticated user one sends a message to user two, a new message object is added to the database. The function first finds if the two users exist. It then constructs a message object having current timestamp, two user ids, and the message body. This message is pushed to the messages collection.

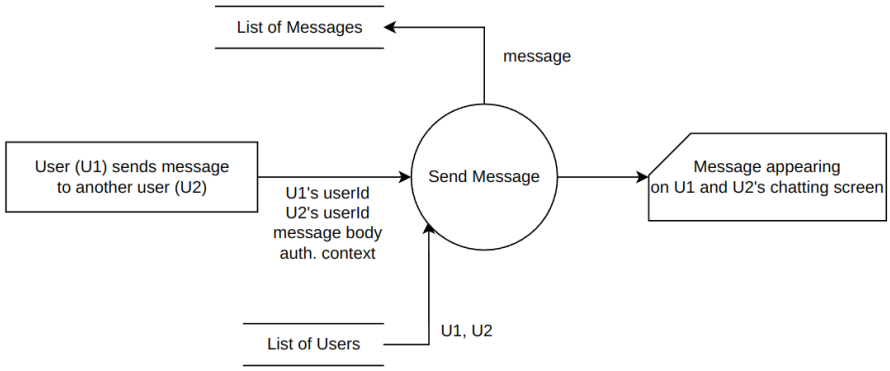
The user interface responds by adding that message right above the typing bar, below all other messages.



Description:

When an authenticated user updates an existing message, the function will fetch that message according to the message id. It will overwrite the old message will the changed message.

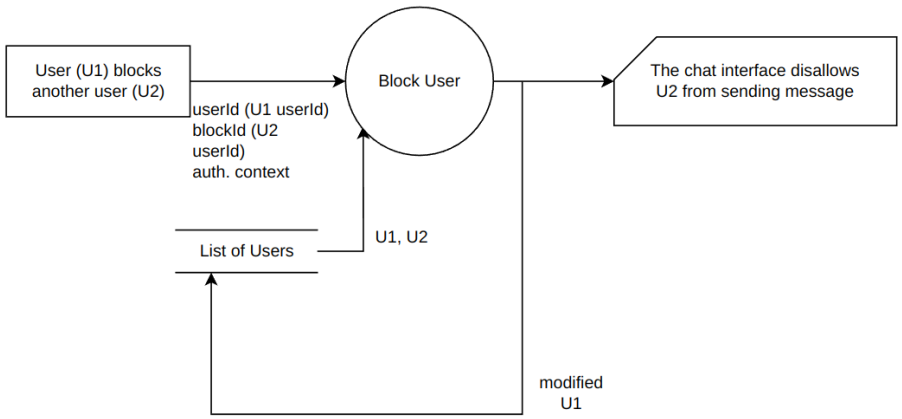
The user interface will change the message body of both users.

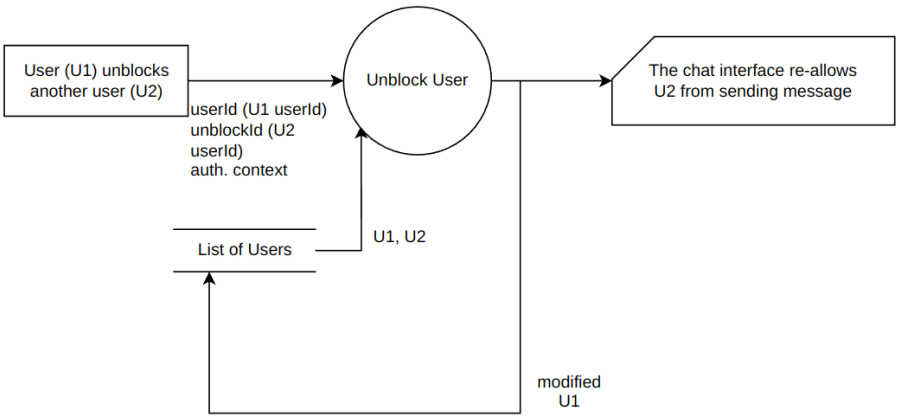


Description:

This function will delete a message according to the message id. It will issue a delete request to the database.

The user interface removes that message from the chat screen and pushes all the messages behind it up one slot.



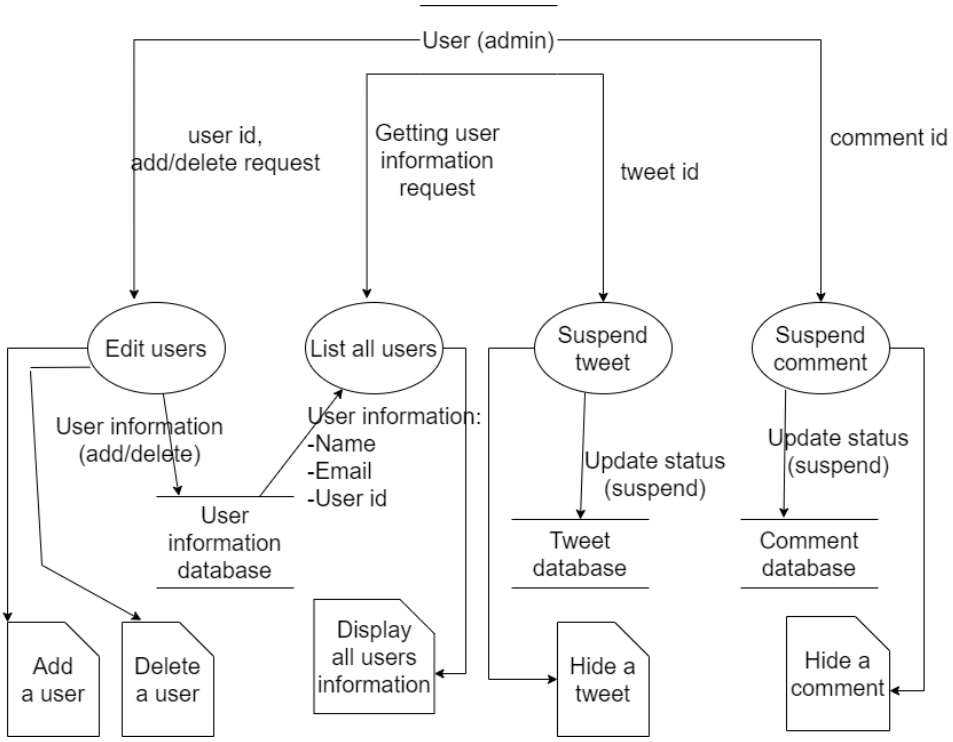


Description:

Both block and unblock work similarly. The authenticated user provides the block id for the function to act on. Both functions will fetch the authenticated user and the user to be blocked. Block function adds that blocked user to the blocked list of the authenticated user, while unblock function removes that blocked user from the blocked list. The modified user is stored back in the database.

If the user is blocked, he will see a disabled typing bar and not be able to send any messages to the authenticated user. If he is unblocked, the typing bar will be enabled again for him to send messages.

2.14. Admin Related



The above data flow diagram includes four functions namely edit users, list all users suspend tweet and suspend comment.

2.15. Suspend Comment

The admin user will provide the comment id for the function. The function will search for the comment in comment database and update its status to suspend. Then the server will not show that suspend comment to the user in the tweet.

2.16. Suspend Tweet

The admin user will provide the tweet id for the function. The function will search for the tweet in tweet database and update its status to suspend. Then the server will not show that suspend tweet to the user.

2.17. List all users

The admin sends a get request to get the list of all users. The function will get all the user information such as name, email and user id from the database. Then the function will send/display the list of all user information to the admin user.

2.18. Edit users

The admin users provide the user id and issue either add or delete request for the function. Then the data in the user information database will be updated. For add request, it will add a new user in the user information database. For delete request, the user will be deleted in the user information database according to the user id that the admin user provides.