General Overview:

This project is an easy social media platform that allows new or existing users to interact with others. We set up an easy login screen with:

L - Login in with your user ID and password

R - Register a new account

Q - Quit the program immediately

L is for existing users; new users have to finish filling in some of their personal information to register.

After logging into our system, the users will see the main menu which contains the name of the user, the latest tweets of the user, and 5 different functionalities including logout.

ST - Search for tweets

SU - Search for users

C - Compose a tweet

L - List followers

Q - Logout

To access different operations, users have to input the corresponding letters. Users can search for tweets by entering keywords, separated by commas. Search results are displayed from latest to oldest, allowing users to browse, inspect, and interact with tweets. Results are paginated for ease of navigation, with options to move forward or backward through pages.

Users can search for other users by entering a keyword matching a username. Results are sorted by ascending name length and displayed with user IDs. Users can select a specific user to view their profile, including statistics, recent tweets, and options to follow or see more tweets.

Users can create new tweets and include hashtags if desired. Tweets are saved and immediately added. The followers list shows all users following the current user. Selecting a follower provides details such as their statistics and recent tweets. Users can also follow back directly from this menu.

Once the user finishes, they can choose the logout option which will return them back to the login screen.

Detailed Design:

Compose Tweet - Gurleen Bamrah: The compose_tweet function is responsible for enabling the user to create and post a new tweet. The user is prompted to input the tweet text, which is then processed to get any hashtags included in the tweet by splitting the words. Then the function checks if any words in the tweet text start with the # symbol and adds them to a set which makes sure that there are no duplicate tweets. The function then generates a unique tweet ID by calling generate_tid function. Then the tweet is stored in the database, first inserted in the tweets table and then the hashtag in the hastag_mentions table linking the tweet's ID.

List Followers - Gurleen Bamrah: The list_followers function gets and displays the followers of a given user. It first uses the follows table to get the users who are following the currently logged-in user, identified by uid. Then displays the followers names and emails showing up to 5

followers per page. If the user has more than 5 followers, they are asked to either view more or quit. When a user selects a follower by number, the function gets and displays more details about that follower, including their number of tweets, the number of users they follow and the number of followers they have. The user can also choose to follow the selected follower or view their tweets.

Log in and Log out - Zhexuan Dai: Log in function asks for users to input a unique user id corresponding to users and a password they set. It then checks the input with the database. Either show log in successfully and proceed to the main user page. Log out simply break the current while loop of login and return to the main page.

Search for Users - Mandi Li: Since it is the searching part, it prompts the user to input a search keyword, which can match usernames in the database. Then check the validity of the input, to make sure the input is alphanumeric and appropriately sized. Once it is validated, search and save the user into the query, and sorted by name length in ascending order. Display the ID and the username of the result, and separate the result into different pages by maximum display number per page. After selecting a user, the number of tweets, followers, and followed users are displayed, along with the three most recent tweets. Users can also browse more tweets or follow the selected user directly.

Search for Tweets - Zikai Zhang: The display_tweet_stats function displays the statistical information of a specified tweet, including the number of retweets and replies. It obtains the number of retweets and replies for the corresponding tweet by querying the retweets and tweets tables in the database and prints the results. The search_tweets function allows users to search for tweets by keyword. Users can input a keyword to search, and the results are sorted by date and displayed on pages. Users can use the page navigation commands to view tweet lists on the different pages, and they can also choose to view the statistical information, reply to tweets, or retweet the tweets.

Display tweets - Zhexuan Dai: The function first call get_tweets_and_retweets to group tweets and retweets together by user id and order by date and time. Display tweets and display each of them in a formatted way.

Workflow - Zhexuan Dai: We struct our interface as 2 while loop, within each while loop, there's some operations corresponding to main page (login, register, quit) and user main page(search tweets, user, compose tweets, list followers and quit to main page)

Test Strategy:

Group testing strategy: Used the sample database given on eclass and added our own test data to it and tested each python file separately first and then when all combined.

Log in - Zhexuan Dai: We tested with different user inputs that the program might counter e.g. enter letters for phone numbers. We added a verification step to ensure the user enters correct information. We also give users the option to enter q to go back to the main page instead of trapping in while loop for log in.

Search for Tweets - Zikai Zhang: We tried searching for tweets with different keywords that have or have no hashtags. The program firstly returns nothing when we enter a keyword. We identify the problem and fix it by lowering the letter case in hashtag_metions table. We also encounter the missing replyto_tid when we reply to a tweet, we fix it by directly call compose_tweet from compose_tweet.py by Gurleen Bamrah.

Compose Tweet - Gurleen Bamrah: Used the database and inserted test data then tested the function compose_tweet with test values. If it worked properly then the values of the tweets table should have been updated and been added to the database. Tested that the hashtags are correctly parsed and inserted. Tested the interaction between compose_tweet() and insert_mentions() function, checks if hashtags are being inserted correctly. One test case was to verify that users can compose tweets, and hashtags are stored in hashtag_mentions. While testing all of these we found 1 bug that duplicate hashtags were inserted in the hashtag_mentions table and I fixed this by using a set to store hashtags.

List Followers - Gurleen Bamrah: Used the database and inserted test data then tested the list_all_followers with test values. We tested if followers were retrieved correctly, if 5 followers show up at a time, verify that the user can select a follower by number and view details. While testing this we found 1 bug that the SQL named parameters were inputted incorrectly causing the program to crash.

Search for Users - Mandi Li: Using the wrong attribute name for the query, after checking the eclass page again, fixed the problem.

Lister Followers - Mandi Li: When the user selects to show more followers but the number of followers is not enough to display more, the program will automatically return to the main menu selection screen. After the change, if the number is insufficient, it will inform the user that there are no more followers and stay on the display screen.

Group Work Breakdown:

Method of coordination: First meeting was done in person on November 7, 2024. This meeting was used to split up the work as follows:

- Login Screen - Zhexuan Dai, Time Spent: 2 hours, Progress: Complete
- Search Tweets - Zikai Zhang, Time Spent: 5 hours, Progress: Complete
- Search Users - Mandi Li, Time Spent: 4 hours, Progress: Complete
- Compose Tweets - Gurleen Bamrah, Time Spent: 4 hours, Progress: Complete
- List Followers - Gurleen Bamrah, Time Spent: 4 hours, Progress: Complete

- Log Out - Zhexuan Dai, Time Spent: 30 minutes, Progress: Complete

We met up on November 14, 2024, on Google Meet to figure out how to combine all our files and created this document as well. In the first group meeting, we tested all operations using the sample database and confirmed they worked correctly. However, we faced difficulties during integration because the original login and logout sections were too complex. And if we want to match the style of login, logout part the workload for other operations was too large. We decided to give up on the original login interface, switched to a new version, and added integrated operations.

The third meeting is on November 15, 2024, on Google Meet. We use the interface.py to check our workflow, find some small bugs in the coding, and change it immediately. After completing the workflow checking, we start finishing the documentation. During the documentation period, suddenly found that the hashtag could not be searched. Fixed it immediately.