Mini Project 1

General overview:
This system allows users to log in or sign up, and then be given a choice between viewing tweets or retweets of other users they follow. Afterwards, the system lists a bunch of system functionalities that the user can select. These include searching for tweets using keyword(s) and see stats based on the selected tweet while also allowing to retweet or reply to the tweet, search for users using a keyword and see stats and the 3 most recent tweets about the selected users while giving the option to follow the user or see more tweets, composing a tweet, displaying the user's followers while allowing them to select the followers and see more stats and 3 most recent tweets while giving the option to follow the follower or see more tweets.

User guide :

- **At the first menu** users can Enter '1' to log in, '2' to sign up or '3' to exit.
- **Logging in** - the user will be prompted to type in their ID and password
- **Signing Up** - users will be prompted to enter in their name, email, timezone, city, and password(mandatory) one by one. The system will display your new user_ID.(don't forget it!!)
- **Main Menu** (after signing in) - user can Enter '1' to see their tweets/retweets, Enter '2' for system functions, or Enter '3' to logout or return to the previous menu (this option is given with all menus)
- **In System Functions** - displays the functions they can run including the following:
    - See below for searching both users and tweets:
    - Enter '3' to compose a tweet - type in your tweet then press enter
    - Enter '4' to see who follows you
    - Enter '5' to log out or '6' to return to main menu
- **Searching a keyword** - this will prompt a user to enter a keyword that may include hashtags, digits, letters and other special characters
    - Search for tweets – displays tweets that have the corresponding keyword in the tweet.
    - Search for users - displays users that have the keyword in their name or city.
- **Navigating through pages:**
    - When Tweets or Users are displayed users are prompted to go to 'next' page, 'previous' page, select a tweet(type in the tweet ID) or quit(quit will bring the user back to the list of system functions)
    - To make a choice, enter the one-word options (next, previous) or enter the number (id) beside the user's name (note: 'select' is not a choice)
    - If the action can't be completed, a message will appear followed by the same output and prompt as before (read carefully for any messages).
- **Viewing a user's profile**
    - When viewing a profile user's may choose to follow that user (type 1), or see more of their tweets (type 2) or leave (type q). See *navigating through pages* for more information on the second option.

Design of the software

Login_screen:

User selects to login or sign up.

registered_users() –This function prompts the user to type in the user_id and password. Password is not shown(done by importing the getpass function which hides the user input as they type it). Checks if the user_id and password is in the database.

unregistered_users() – This prompts users to type in their name, timezone, email, city, and password. All the inputs are optional EXCEPT for the password. Data is recorded into the data base and then displays the user's NEW user_id.

Once logged in, the user has the option to see tweets/retweets, other functions, logout and exit. In the first option, user has the option to select separately whether they want to see tweets or retweets. In case of tweets, user is given the option to see the next page or the previous page of tweets AND the ability to select a tweet by their TID if it exists in the display. The display part and getting the TID of the selected tweet is handled inside the "display_stuff()" function.

DIsplay_stuff() – Parameters include the list, num of items in the list to display at a time, true/false(1/0) condition for if selecting an item on display is needed or not. Essentially, use list index slicing to display the items in the list, and in case the last parameter is true, check if the selected id is in the list. IF the list is empty , print("no items to be displayed")

Once back to the main menu, users can select the second option to use system functionalities.

search_users():

This function allows a user logged into Twitter to search for other users based on a keyword. It queries the database to find users whose names or cities contain the entered keyword. The results are then displayed, with users whose names contain the keyword listed first, sorted by name length, followed by users with the keyword in their city, sorted by city name length.The function **displays 5 users at a tim**e and gives the option to **view the next/previous** 5 users, view a user's profile, or continue. If a user ID is entered, the corresponding profile is opened, showing the user's ID, name, number of tweets posted, three most recent tweets, and follower/following count.Once a user's profile is open, there is an option to follow that user. If chosen, this action is recorded in the database. Tweets are also displayed 5 at a time, with options to view the next/previous set or quit.

compose_tweets():

The user is prompted to input a text to add to the tweet table. The function checks to see if the tweet's table is empty before inserting the text into the table. If the tweet table is empty, that means that there is no tid, therefore, this tweet would be the first tweet in the tweet table. Otherwise, the function takes the maximum value for tid and adds one to it which will be associated with the text the user just typed. Since tweets can be used to reply to another user, the function checks to see if it is being used to reply to someone or not. Either way, it correctly adds the text along with tid, date, and other values in their respective columns to the tweet table. After adding the text into the tweet table, the function checks to see if there are any hashtags in the text. Since hashtags must be inserted separately from one another, the function accurately separates them, checks to see if it already exists and adds them to a list which will be iterated to add into the hashtag and mention table.

list_followers():

This function prints a list of followers that follows the user. This is done by calling a function called list_of_followers() which returns a list of followers that includes their user id as well as their name. The function then prompts the user to select a follower to see more information such as the number of tweets via a count_tweets(), number of followers via count_followers and list of tweets made by the follower via list_of_tweets(). The function then displays up to 3 tweets done by the follower and asks the user if the user wants to follow the follower back. The function checks to see if the user is already following the follower and exits the function loop if the user is already following the follower, otherwise, the function adds a new value to the follow table.

search_for_tweets():

The user is prompted to input keywords to use for searching through tweets ordered by most recent tweets. This will work with multiple keywords and these keywords will be used in the query to search for similar tweets. Keywords starting with # will be searched as hashtags, and keywords not beginning with # will be searched generally. Up to 5 tweets that match the description are then displayed. If there are more than 5 tweets, the user is then prompted to select whether they want to see more tweets. When quit is selected, the user is prompted to input the tweet id of a specific tweet that they want to operate on. The input is checked whether the id is valid or not. The user is then prompted to indicate if they want to reply to this tweet or retweet this tweet. Here compose_tweet() and retweeting() functions are used.

Testing strategy

- Created a mini interface that bypasses some questionnaires to allow us to test our system functions quicker.
- Created a small database with which we can use to test our system function's queries on. The schema of the database is based on what is provided in eclass.
- Hard-coded some values during our test with the purpose of testing edge cases to come up with a solution to certain problems.
- Specifically related to user_input testing: essentially, checked to make sure the user_input accepts ONLY the given options and doesn't crash incase of a valid input.

Scenarios:

- We tested all the choices given my prompts with combinations of special characters/letters/digits, ensuring to display the right messages and handle corresponding errors
- Tested for when the user attempts to follows themself
- Tested for multiple hashtags in an entered keyword
- Tested to check if the user is already following someone. If so, let them know they are and not allow it.
- Tested to not allow retweets of the same tweet.
- Tested to allow multiple replies to the same tweet
- Tested to see if the item the user selects from the display actually exists in the given display
- Tested if the user is new or doesn't have followers, display no followers yet and then exit the function to the menu.

Group Work Break-down Strategy:

- We distributed the Login Screen and System Functionalities assuming the length of the specifications descriptions reflected its workload. We regularly updated each other over discord about any changes we made. Common functions needed were determined before starting the project and chosen by members as they were needed for that member's system function.

AKILA - error checking the database path to make sure it exists(~30 mins)

- login screen and the other functionalities needed inside it(including registering new users and giving a new id , displaying tweets and retweets, allow users to select items on the displayed tweets) (~4 hours including testing)
- Common Functions - finding num_retweets(), finding num_replies(), display_stuff(), check_select(), retweeting()(~4 hours including testing)
- final testing(including running through all the system functions the group members worked on, fixing any errors that pop up on other members code, implementing missing functionalities) AND edge case testing(~6 hours)
- beautifying the results getting printed so that it is more readable on the terminal(~1.5 hours)
- documentation(~1 hour)

MELODY - Search for tweets functionality (~2h)

- Common Functions - tweet_stats (~10 minutes)
- documentation(~1 hour)

ALISSA - Search for users functionality (~6 hours)

- Common Functions - count_following(), count_followers(), count_tweets(), follow() (~2 hours)
- documentation(~3 hour)

AIVAN - Compose a tweet and List followers functionalities, Creating database for tests

- Common functions - list_of_tweets(~3 hours), list_of_followers(~4 hours)
- documentation(~1 hour)