CMPUT 291 Mini-Project 1 Design Doc

General Overview

Our application represents a very rudimentary social network, akin to X (formerly Twitter). Users can log in using their User ID and password.

From the main menu, users are shown their followed users most recent tweets. Users can input one of the following options:

- P: See their followed users 5 previous Tweets
- N: See their followed users 5 next Tweets
- [id]: Select tweet to get more options
- T: Search for tweets by inputting keywords
- S: Show their followed user's 5 most recent tweets
- C: Compose a tweet
- U: Search users using a single keyword.
- F: Lists all users following the current user
- L: Logs current user out and returns to Login menu
- E: Exits the application

From the Tweet menu, users can input one of the following options:

- S: Get the stats of selected Tweet (Reply count and Retweet count) and is returned to the main menu
- R: Reply to the selected Tweet
- T: Retweet the selected Tweet
- B: Return to the main menu

From the Followers menu, users can input one of the following options:

- [num]: Displays the user at index num's Tweet count, Follower count, Following count, up to 3 of their most recent Tweets
- N: Display the user's next 3 Tweets
- P: Displays the user's previous 3 Tweets
- B: Return to the main menu

When a user searches for a Tweet, they can input one or more keywords and will return all Tweets that have at least one of the keywords in the Tweet text or a mention (#). Results are displayed 5 at a time and ordered from newest to oldest. Users can then reply to or retweet any Tweets found, as well as see the stats of a selected Tweet.

When a user searches for another user, they can only enter a single keyword and return all users who either have that keyword in their name or city. Results are displayed 5 at a time and ordered by name length, then city length. The user can then see the next/previous 5 users, select a user to see their stats or return to the main menu.

Software Design
Our application was designed and developed using Python to handle the main functionality and SQLite3 to handle our database. We chose both Python and SQLite3 for their ease of use, their content covered in labs, and each member's familiarity with the languages.

Our **main.py** handles our main application logic, database connection, login menu, and our homepage. Through the main() function, input file redirection is verified, then the application connects to the provided database and the main logic for the application begins.

loginMenu() is then called where the user is prompted to log in, register, or exit. Input validation is conducted and the program moves to validateUser(), registerUser(), or the program exits. Once registered and/or verified and logged in, we then move to the homepage().

From the home page, we display the 5 most recent tweets or retweets from the current user's followed users. After prompting the user, input validation is again used to either display more tweets, create a tweet via composeTweet(), search tweets via searchTweets(), display followed user tweets via getTweets(), search users via userSearch(), display followers of current user via getFollowers(), logout and return to login menu, or exit program.

tweets.py contains the logic for getting, finding stats for, composing, searching for, replying to, and retweeting tweets.

- getTweets() uses a query that uses a union of joint tables that returns an ordered list of tuples, containing all tweets and retweets for a selected user.
- searchTweets() finds tweets that match at least one keyword in a user-provided list. It will check in the mentions table if the word begins with '#', else, it checks tweets
- composeTweet() allows user to create a tweet, then updates the tweets table with the tweet and a new unique tid.

follows.py contains all functions that deal with follower stats, and following a new user

user_info.py contains functions related to user searching. The query involved checks whether the keyword is LIKE a user's name, OR a user's city, then orders the results by the length of the name, then the length of the city.

printing.py contains all functions to display tweets, followers, and stats.

authentication.py contains validateUser() - which will mask password input and validate the password inputted against the user's id - and registerUser(), which prompts user to input pertinent information. It also uses input validation as well as checks for SQL errors when updating the users table in our database.

Testing Strategy

Out testing strategy was relatively simple. Most of the code was written on the lab machines themselves, with our final project being fully tested on the lab machines. We tested our queries using a modified version of test1.DB from Assignment two. This allowed us to test the accuracy of our queries, and we feel they are resilient.

Groupwork Breakdown

David: Login page, main application logic, and cleanup/organization. 14 hours
Ian: Tweet search, user search, follows list. 16 hours
Ankriti: Compose tweet. 2 hours

We used GitHub to mainly keep track of our project, using branches for different elements. We also used Discord to communicate with one another and keep the project moving.