

General Overview of the System:

Our project is a system that manages user data and interactions within a database. The system provides an interface for accessing social media functionalities such as viewing a feed, composing tweets/retweets, searching for tweets, and managing user connections (following).

System Workflow:

1. Database Connection: When launching the program, users are prompted to enter the filename of the database they wish to use. Our system will then connect to the specified database, to ensure the correct scheme is in place
2. Login/Register Screen: Users can choose the "login" option if they have an existing account, or the "Register" option to become a user. Once successfully logged, users are directed to the main screen
3. Main Screen functionalities:
 - a. View Feed (Displays the first 5 tweets + retweets of users being followed)
 - b. Compose Tweet (create a tweet/retweet)
 - c. Search Tweets(Allows users to search for specific tweets using keywords. The results show detailed information about the tweet, including the content, date/time, and the user ID of the author)
 - d. List Followers (Lists all users who follow the logged-in user)
 - e. Follow a User (Enables user to follow another user by entering their user ID)
 - f. Unfollow a User (Enables user to unfollow a selected user)
 - g. Logout (Logs user out and returns to the initial Login/Register screen)

User Guide:

1. Ensure user has appropriate SQLite database file prepared, and launch the program by entering the filename of the database to establish a connection
2. Once successful, Login as an existing user or Register as a new user by providing your name, email, phone number, and create a new password
3. Once logged in, you can browse through a variety of options which include:
 - i. View Feed: click on "View Feed" to see the most recent tweets and retweets from the accounts you follow
 - ii. Compose Tweet: Use "Compose Tweet" to write and publish a new tweet or a retweet
 - iii. Search Tweets: Select "Search Tweets" to find tweets by entering a keyword
 - iv. List Followers: View the list of users who are currently following you
 - v. Follow/Unfollow: Input the user ID to follow or unfollow another user
 - vi. Logout: Select "Logout: to safely end your session"

Component Structure and Design:

main.py: Main Application Logic (manages database connections, user interactions, and tweet-related functionalities)

- a. Connect_db: establishes a connection between the SQLite database and enables foreign keys. The database name is not hardcoded and must be passed through by the user.
- b. Is_valid_input: Validates user inputs for email and phone formats
- c. Register_user: Registers a new user by taking their details, validating them, and inserting them into the users table in the database. If the user already exists within the database, they will get an error message "Error: Email or phone already registered"
- d. Login_user: Authenticates a user by checking their user ID and password. If the system does not recognize the user, they will get an error message "Invalid User ID or Password."
- e. Display_feed: Fetches up to 5 tweets and retweets from followed accounts.
- f. Compose_tweet: Allows users to post new tweets, including replies. It inserts the tweet into the tweets table, capturing its timestamp, content, and reply reference
- g. Search_tweets: Searches tweets by keywords
- h. Search_users: Searches for users
- i. List_followers: Lists all followers of logged-in user
- j. Follow_user: Enables user to follow another user by their user ID
- k. Unfollow_user: Enables users to unfollow another user by their user ID

Testing Strategy:

Our testing strategy focused on ensuring comprehensive coverage of both functional and edge case scenarios across all features of the project. The strategy was divided into unit testing, integration testing, and system testing phases. For unit testing, each function was tested with a variety of inputs to confirm expected outputs, including some invalid input to check the error message. Integration testing ensured interactions between different functions, such as register_user, compose_tweet, list_followers and so on. System testing is simulated real-world workflows, such as registration as a user and login to check. The test covers all database queries, ensuring it works properly.

Group Work Break-Down Strategy:

Angela: Implemented the following queries: 1) Find out if userID and pwd exist in table by searching through it 2) Register any users that do not exist 3) Search for tweets by prompting users to enter one or more keywords. If the tweet matches a keyword or a hashtag, return its content with the unique identifier, the author ID, the content of the tweet, date, time, and if the tweet was a reply to another tweet, it will store the original tweets unique ID. 4) Compose a tweet: prompting the user to compose a tweet 5) List followers .

I also assisted in testing functions via unit tests. As well, I created the DesignDoc, where I worked on the General Overview of the System, including the System Workflow and User Guide, and the Component Structure and Design.

Hana - focused on building the core features for user registration, login, tweet composition, and displaying a personalized feed. I set up validation for registration inputs (like email format and password confirmation) to improve data integrity. For the Feed Display, I implemented logic to show tweets and retweets from followed users in chronological order, with a "load more" option. I also made sure SQL queries were secure and helped troubleshoot user input and data validation issues. Throughout, I worked with the team to test and integrate these features smoothly, addressing any issues that arose along the way.

Sho - worked on the sql. write the sql for find most 5 tweets and retweets for every login user, list user by keywords, find the detail of user after another user clicked into their profile, list 3 recent tweets of that user, add the follower to user's follower list and able to check more tweets after click check more tweets. I also helped team with using github and test the program at the end, also discuss with teammates about the functionality implement ways. Throughout this process, I collaborated with the team to integrate these features with the overall application, assisting teammates with overlapping functionalities and resolving any database conflicts.

Hooriya - worked on the functionality for searching tweets, searching users, listing followers, following a user, and unfollowing a user. For the ****Search Tweets**** and ****Search Users**** features, I implemented the logic to filter and retrieve relevant data based on user input. I also developed functionality to ****list followers**** and enabled users to follow and unfollow other users, with appropriate validations to

prevent redundant actions. After implementing these features, I conducted extensive testing to ensure they worked as expected, verifying edge cases like searching for non-existent tweets or users, and handling invalid follow/unfollow attempts. I also helped debug issues related to database queries and user input errors. Throughout this process, I collaborated with the team to integrate these features with the overall application, assisting teammates with overlapping functionalities and resolving any database conflicts.