Mini Project 1 Report

Group Members:
Mingwei Li
Shuyang Li
Xuanyi Wu

1. General Overview
When the program runs, user will see a login screen first. The user can login or register or exit. If the user choose exit, the program will shut down.

After the login or register steps, user will go to the main screen and see 5 tweets of their followee. Then, the program will ask the user to take further action. The user can choose to:

1. Show more tweets of their followee
2. Show information of one tweet
3. Search for tweets
4. Search for users
5. Compose a tweet
6. List user's followers
7. Manage user's lists
8. Logout

Every further action will call certain function in the program, and after the user finished their further action, they can come back to the main screen and do other further actions. If the user choose to logout, he/she will go back to the login screen.

2. A Detailed Design
1.connect to sqlplus in main function
2.LogIn()
  login for both registered and unregistered users
3.tweets_retweets(uid)
  after login, the user will see the tweets and retweets posted by his/her followers (5 at a time). The program also asks user to do more actions:

        3.1. More tweets (call: list_nterms)
        3.2. One tweet information (call: tweet_info)
                3.2.1 Reply: reply a tweet (call: compose)
                3.2.2 Retweet: retweet a tweet
        3.3. Search for tweets (call: search_tweets)
          give the tweets containing the keywords (5 at a time)
                3.3.1. Show more tweets (call: list_nterms)
                3.3.2. See more information of a tweet (call: tweet_info)
                3.3.3. Reply a tweet (call: compose)
                3.3.4. Retweet a tweet
        3.4.Search for users (call: search_users)
          give the users containing the keywords (5 at a time)
                3.4.1. Show more users (call: list_nterms)
                3.4.2. See more information of one user (call: user_info)
        3.5. Compose a tweet (call: compose)
        3.6. List followers (call: list_follwes)
        3.7. Manage lists (call: manage_list)
        3.8. Logout

4.list_nterms(rows,jth, num_t)
        show every 5 (or 3) results a time that satisfies the condition

return False if there is no more tweets to be shown

5.tweet_info(stweet)

gives the number of retweets and the number of of a tweet

6.compose(uid, reply, reply_to)

compose a tweet(or reply), add the hashtags to the mentions and hashtags if needed

7.search_tweets(words)

get the tweets containing one more keywords(both in hashtags and text)

8.search_users(keyword)

get the user containing the keyword(both in name and city), first show users containing keyword in name (length of name short first), then show users containing keyword in city(length of city short first)

9.user_info(uid, selected_uid)

show the number of tweets, the number of users being followed and the number of followers

9.1 Follow this guy

9.2.Show more tweets (call: list_nterms)

10.list_follwes(uid)

list the user's followers and the user can choose one follower to see his/her information
(call: user_info and can also do the follow and see more tweets)

11.manage_list(uid)

11.a. get a listing of your lists (call: get_list)

11.b. See the lists that you are on (call: see_list_enrolled)

11.c. Create a list (call: create_list)

11.d. Add a member from your list (call: add_members)

11.e. Delete a member from your list (call: delete_member)

11.f. back to menu

12. get_list(uid)

return the lists' names created by the user

13. see_list_enrolled(uid)

return the lists' names where the user is included

14. create_list(uid)

user create a list

15. add_members(uid)

add somebody to the list created by the user

16. delete_member(uid)

delete somebody from the list created by the user

3. The Testing Strategy

We test our program in two aspects: check user input and check with data written by ourselves.

In order to check the user input, we input some invalid input such as inputting character when they need integer and inputting integer when they need character. We also check the input integer whether it is in the range of program provided.

In order to test our program, we wrote the data including some extremely cases. For example, some users' name may be same as the cities' name. We need to except that result in the searching cities' name. we also check the tweet if the number of tweets is less than 5 when they need to print 5 tweets in one time.

The number of bugs we found and nature of those bugs:
Invalid user input: there are almost 8 bugs and the nature of those bugs is that program would crash by invalid input.

Logical bugs: there are almost 4 bugs and the nature of those bugs is that program would crash by infinite loop.

Syntax error: there are almost 5 bugs and the nature of those bugs is that program would crash by some invalid statements.

4. The group work break-down strategy.
Firstly, we read the question together and discuss what function we would write and how we should break down. This is our group work strategy:
Mingwe Li:
compose(),manage_list(),get_list(),see_list_enrolled(),create_list(),add_members, delete_member()
(Time estimate:20h)
Shuyang Li:
user_info(),search_tweets(),search_users(),list_follwes()
(Time estimate:20h)
Xuanyi Wu:
LogIn(),tweets_retweets(),tweet_info(),list_nterms(), (Time estimate:20h)

We use the email and face to face discussion to make sure the project on track.

The decision beyond required:

1. Search tweets we use case insensitive
2. Check the user input
3. Avoid the user following himself/herself