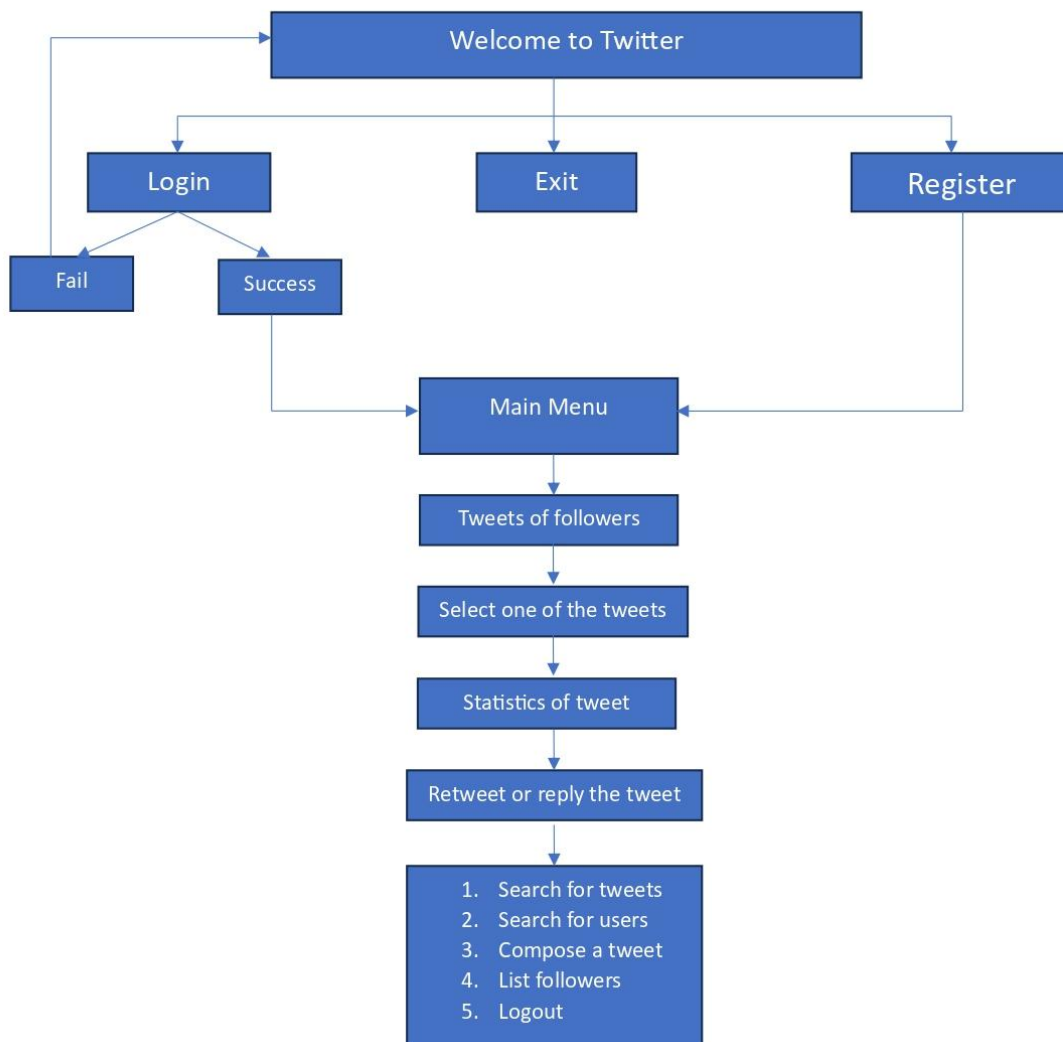


Design Document

Team Byte: Dhruv Kairon, Jaspreet Singh Chhabra, Rithwik Korukonda Bhattar

General Overview

Our program includes one Python file main.py.



User Guide

This guide entails how to traverse through the program and contains all the functionality.

This is the initial landing page where the user can choose to login, register, or exit the program.

```
jchhabr1@ug01:~/CMPUT291/miniproject1>python3 finalapp.py
*****
Welcome to Twitter!
*****

Please select an option:
1. Login
2. Register
3. Exit
Enter option: 
```

If the user chooses to login, the program displays 5 tweets or retweets from the users followed by the user at a time. The user is given a choice to view the remaining tweets or retweets if there are more than 5.

```
1. Login
2. Register
3. Exit
Enter option: 1
*****
Login page
*****

User ID: 1
Password: password1

Welcome, User1

*****
Main menu
*****

Tweets

Tweet ID: 2
Writer: 2
Date: 2023-01-02
Text: Hello, Twitter!
Reply to: None

Tweet ID: 4
Writer: 3
Date: 2020-01-04
Text: Retweeting tweet 2.
Reply to: None

Tweet ID: 5
Writer: 2
Date: 2019-01-04
Text: Retweeting tweet 2.
Reply to: None

Do you want to select a tweet? (y/n)
Enter option: 
```

If the user is not registered or invalid credentials are inputted, the user is asked to register.

```
Please select an option:
1. Login
2. Register
3. Exit
Enter option: 1
*****

Login page
*****

User ID: 1230
Password: doesnotexist
Invalid username or password
Do you want to register? (y/n)
Enter option: █
```

If the user chooses to register, they can by selecting option 2.

```
*****
Welcome to Twitter!
*****

Please select an option:
1. Login
2. Register
3. Exit
Enter option: 2
*****

Register page
*****

Name: John Doe
Email: example@gmail.com
City: Edmonton
Timezone: 2.5
Password: password2
Welcome, John Doe
*****

Main menu
*****
```

Once the user is successfully logged in, they can select a tweet or retweet which is being displayed.

```
Do you want to select a tweet? (y/n)
Enter option: y
Select a tweet to display the number of retweets and replies to the tweet.
Enter tweet ID: 2
Number of retweets: 0
Number of replies: 0

Do you want to retweet or reply to your selected tweet? (y/n)
Enter option: y

1. Retweet
2. Reply

Enter option: 1
Retweet successful!
```

The user is given a choice to either retweet the tweet or enter a reply to the tweet.

```
Do you want to select a tweet? (y/n)
Enter option: y
Select a tweet to display the number of retweets and replies to the tweet.
Enter tweet ID: 2
Number of retweets: 1
Number of replies: 0

Do you want to retweet or reply to your selected tweet? (y/n)
Enter option: y

1. Retweet
2. Reply

Enter option: 2
Enter reply: I am replying to your tweet!
Reply successful!
```

Once the user replies or retweets or selects no as the response, the next menu is displayed. The user is given 5 choices: search for tweets, search for users, compose a tweet, list followers and logout. If the user chooses to search for a tweet by entering 1, they are asked to enter keywords they are looking for in a tweet. Then tweets are displayed which have at least one of the keywords. 5 tweets are listed at a time, but the user can choose to see more. The user can choose to view statistics for a tweet: number of replies and number of retweets. Then the user can also either reply or retweet the tweet.

```
Do you want to select a tweet? (y/n)
Enter option: n
Please select an option:
1. Search for tweets
2. Search for users
3. Compose a tweet
4. List followers
5. Logout
Enter option: 1
Please enter one or more keywords: replying retweeting #feelinggood
Tweet ID: 1004
Writer: 1
Date: 2023-11-07
Text: I am replying to your tweet!

Tweet ID: 1005
Writer: 4
Date: 2023-11-07
Text: replying to tweet 4

Tweet ID: 1000
Writer: 7
Date: 2023-10-31
Text: it was a beautiful day bro. #feelinggood

Tweet ID: 3
Writer: 1
Date: 2023-01-03
Text: Replying to tweet 1.

Tweet ID: 4
Writer: 3
Date: 2020-01-04
Text: Retweeting tweet 2.

Displaying 5 of 6 tweets. Do you want to see more? (y/n):
y
Tweet ID: 5
Writer: 2
Date: 2019-01-04
Text: Retweeting tweet 2.

Do you wish to see statistics for a tweet? (y/n): y
Which tweet do you wish to learn more about?
Enter the Tweet ID: 3
Retweets count: 0, Replies count: 0
Do you wish to reply to a tweet or retweet? (reply/retweet/no): reply
Enter the Tweet ID: 5
Compose a reply: replying to a tweet!
Successfully Replied!

Do you wish to reply to a tweet or retweet? (reply/retweet/no): retweet
Enter the Tweet ID: 7
Sucessfully retweeted!
```

If the user opts for searching users by inputting 2, they will be asked to enter a keyword which is then used to display all the user_ID and names whose names or cities matched with the keyword. First, the user_IDs are displayed in ascending order of name length and then user_IDs are displayed which do not have the matching name but the city. Again, the output will be displayed 5 at a time and the user will be given a choice to see more if there are more than 5. Then the user is asked to enter the user_ID of a user whom they want to learn more about. If they select yes, they are displayed the number of tweets, number of users being followed, number of followers and the 3 most recent tweets. The user is given a choice to see more tweets.

```
Please select an option:
1. Search for tweets
2. Search for users
3. Compose a tweet
4. List followers
5. Logout
Enter option: 2
What user would you like to find: User
Matching Users:
User ID: 1
Name: User1
-----
User ID: 2
Name: User2
-----
User ID: 3
Name: User3
-----
Enter the User ID to view details or enter 0 to go to the main menu:
3
Follower Details:
Name: User3
Number of Tweets: 5
Number of Users Being Followed: 4
Number of Followers: 2

Want to see more tweets? (y/n)
Enter option: █
```

Then the user is given an option to follow the user they have selected.

```
Enter the User ID to view details or enter 0 to go to the main menu:
3
Follower Details:
Name: User3
Number of Tweets: 5
Number of Users Being Followed: 4
Number of Followers: 2

Want to see more tweets? (y/n)
Enter option: y
Tweet ID: 13
Writer: 3
Date: 2023-11-07
Text: packing
Reply to: None

Tweet ID: 14
Writer: 3
Date: 2023-11-07
Text: nah wildin
Reply to: None
```

```
Tweet ID: 5
Writer: 3
Date: 2023-11-06
Text: fair enough
Reply to: 3

Next 3 tweets? (y/n)
Enter option: y
Tweet ID: 6
Writer: 3
Date: 2023-11-06
Text: i am swag
Reply to: None

Tweet ID: 4
Writer: 3
Date: 2023-01-04
Text: Retweeting tweet 2.
Reply to: None

Do you want to follow this user? (y/n)
Enter option: y
You are already following this user.
```

If the user selects to compose a tweet by selecting option 3, they are asked for the input.

```
Please select an option:
1. Search for tweets
2. Search for users
3. Compose a tweet
4. List followers
5. Logout
Enter option: 3
Compose a tweet: Good morning twitter!
Successfully Tweeted!
*****
```

If the user selects to list followers by inputting 4. The user is displayed all the users which follow the user. The user is able to select a follower to view more details about them. The user can view the number of followers, number of users being followed.

```
Please select an option:
1. Search for tweets
2. Search for users
3. Compose a tweet
4. List followers
5. Logout
Enter option: 4
Followers:
1. User1 (User ID: 1)
2. User2 (User ID: 2)

Enter the follower number to view details or enter 0 to go to the main menu:
2
Follower Details:
Name: User2
Number of Tweets: 4
Number of Users Being Followed: 4
Number of Followers: 3

Want to see more tweets? (y/n)
Enter option: y
Tweet ID: 10
Writer: 2
Date: 2023-11-07
Text: go oilers
Reply to: None

Tweet ID: 11
Writer: 2
Date: 2023-11-07
Text: koala
Reply to: None

Tweet ID: 12
Writer: 2
Date: 2023-11-07
Text: crazy game
Reply to: None

Next 3 tweets? (y/n)
Enter option: y
Tweet ID: 2
Writer: 2
Date: 2023-01-02
Text: Hello, Twitter!
Reply to: None

Do you want to follow this user? (y/n)
Enter option: y
You are already following this user.
Start? (y/n)
Enter option: n
Goodbye!
```


If the user opts to logout and then can proceed to exit the program.

```
Please select an option:
1. Search for tweets
2. Search for users
3. Compose a tweet
4. List followers
5. Logout
Enter option: 5
*****

Welcome to Twitter!
*****

Please select an option:
1. Login
2. Register
3. Exit
Enter option: 3
Goodbye!
jchhabr1@ug01:~/CMPUT291/miniproject1>
```

Detailed Design

Main functions breakdown:

1. Login
2. Register
3. Populate data
4. Main menu
5. Retweet
6. Compose tweet
7. Searching for tweets
8. Search users
9. List users
10. View follower details
11. Follow user

Login: this function allows the user to login into the database. It asks users for the user_ID and password and then checks if a user exists, it calls the main menu function. If not, it leads the user to the register function.

Register: this function allows the user to register to the database. This function can be used to set up a password and generate a user_ID for the user. It sets the user_ID to one plus the existing maximum user_ID.

Populate data (commented out): this function was used by us to populate our test database. It includes user tweets, retweets, and replies with options to add and delete replies and retweets. Also includes the option to write a tweet and follow people. It was needed to fill our database enough to check over 5 requirements.

Main Menu: This is our primary function or the second menu after logging in. It asks the user to start so it can present the tweets and retweets of the user's followers. After displaying the following tweets and retweets as well as the statistics based on the specifications, the user will get a submenu of the options 1 to 5 specified for searching tweets, users, etc.

Retweets: this function takes user_ID and tweet_ID as arguments and runs a SQL query which inserts a tuple into the retweets table.

Compose tweet: this function allows users to compose a tweet or a reply. It takes two arguments: user_ID and reply_to, Reply_to is an optional argument which is set to None by default unless it is a reply, then it would be the tweet_id of the tweet which the user wants to reply to. If it is not a reply, we insert a tuple into the tweets table after taking user input. But if it is a reply, the tuple which we insert in the tweets table contains an additional column(reply_to). We then make sure to store the last inserted tweet's tweet_id. We then iterate through the input text by user to find all the hashtags and then we insert them into the hashtags table and mentions table. We insert into the mentions table regardless if it is a new hashtag or not but for the hashtags table we use the command insert or ignore to only insert if it is a hashtag which does not exist in the table. Used left join so that all the tuples from tweets are there even if they do not have a matching tuple in mentions, which is followed by two join conditions which look for the keywords and hashtags.

Searching for tweets: this function allows users to search for tweets based on the keywords they provide. This function takes only user_ID as an argument. We first get the user input and turn it into a list of strings using split function (after strip). Then we look for any hashtags in the user input and form two strings: hashtags, which is a concatenated string that includes all the hashtags, and keywords_updated, which is again a concatenated string that includes all the keywords user provided. Both are concatenated in such a way (using join) so that they can be inserted into SQL queries. Then we run the SQL query which gets tweet_id, text, date, replies count and writer. To get replies count and writer, we used nested queries. Once this runs, we have a nested tuple with all the tweets that are matching at least one of the keywords or hashtags in descending order. Then we proceed and display 5 tweets at a time and ask user if they want to see more. Then ask the user if they would like to know the reply count or retweet count of a tweet, which we already have in the tuple. Then ask the user if they would like to reply, retweet or quit. If retweet: call the retweet function and if reply: call the compose tweet function.

Search for Users: We first need a query to get the users that contain the keyword or the cities that contain the keyword. Then we get the list through list comprehension of user names that contain the keywords and then user cities that contain the keyword but not the user name. After that, using loops, we print the details in batches of 5. After this, we ask the user if they want to see more details or not, and if they do, we call the function `view_follower_details` to get the details of the user using multiple queries, and then ask if the user wants to see more tweets and wants to follow. If they want to follow, we use a `insert into` query to add the follow, and if they want to see more tweets, we just use list indexing to get the rest of the tweets. Throughout this function, the user also has the option to go back to the main menu.

List Followers: We first need a query to list all the users that follow the selected user, so we can get `usr` and `user_name`. Then if the user wants to show the details, we need another query from the `users` table to get the details, and if the information is valid, we need multiple sub queries to get the number of tweets, number of users the follower is following, number of followers the follower has, and the three recent tweets which we can get by getting all the tweets by order of date and indexing the recent 3. If the user wants to see more tweets, we can just index it again such that it shows everything except the recent 3 tweets. If the user wants to follow that follower, we can write a `INSERT INTO` query to do the same. Also, in each of these scenarios, we will provide the user the option to return to the main page.

Follow_User: We have an `insert into` query to insert the follow to the specified user, and the code also checks whether the current user already follows that user or not.

View_Follower_Details: This function uses multiple sql queries to get the details of the respective user, and gives the user the option to follow, which then goes to the `follow_user` function, or see more tweets, which just indexes the tweets list to get the rest of the tweets.

Testing strategy

We mainly relied on a custom test database to test our data, checking the database to see if our results are accurate. Additionally, we made sure to have a lot of print statements in our code whenever we want to find a mistake in the code. We also asked chatGPT to give us some sample data in order to test our function, and then made a new function `populate_data`, to add all the data we want. We all read the spec document carefully so that we would have a project that adheres to all the requirements. We manipulated the data so that it would go through all the different scenarios for example, to make sure that while searching for tweets even though the tweet does not contain a hashtag, the `mention` table might contain a hashtag under the same `tweet_ID`. While testing we found plenty of bugs. Here are some examples:

1. The code was treating a hashtag with different casing as a different hashtag. To fix this first we got a tuple with all the hashtags from the `hashtags` table and converted it into a list with all the hashtags and used that to compare the hashtags received by the user in

composing a tweet function to make sure that any casing with the same letters it not treated as different by the database.

2. We found multiple bugs while integrating specific functions done by different group members. There were some formatting bugs as everyone had their own formatting outputs for tweets.
3. Another big issue we found was the smoothness of the loop and going back functionalities. The plethora of conditional statements made it difficult to debug
4. There were also issues regarding searching for users and displaying only 3 tweets first and subsequently later upon asking.
5. Edge case errors were also significant as we tried to map unlikely responses and pruned those errors.

Group Work-Breakdown Strategy

To ensure that everyone had similar amount of work to be done, we split up the work:

Dhruv: Menu screens and logout.

Time Spent: 15 hours.

I spent some time making the initial menu screen when the other members were trying to complete the questions 1 to 4 in their own databases and systems. Created an individual database which had the same initial starting data we got from ChatGPT. Made the `populate_data` function to populate it. Finished login, register and a rudimentary inner `main_menu`. After the group finished their versions of the other search functions. I updated the `main_menu` to incorporate their functions. Then we debugged together

Jaspreet: Functions for retweets, searching tweets and composing tweets.

Time spent: 15 hours.

Most of the time I spent doing this project was writing up python code and sql queries for questions 1 and 3. I made sure that all the requirements entailed in the document are fulfilled by these. After getting the initial data from chatGPT, I made my own database to test all the edge cases and used it to find bugs in the program. I also wrote up the user manual.

Rithwik: Functions for Search for Users and List Followers and the respective other functions such as `follow_user` and `view_follower_details` required for those tasks.

Time taken: 15 hours

I spent time coding up the functions for the tasks search for users and list followers. I used sample test data to test my functions separately, handling edge cases and ensuring that it works smoothly. I also contributed towards bringing together all the work which the three of us have done. I also contributed towards writing the design document and the user manual.

Work we did together: Debugging, testing, and integrating individual functions together. We all dissected the problem and made sure functionalities worked individually before putting it all together. Unsurprisingly, there were so many bugs initially when we put everyone's individual functions together but we eventually resolved all issues and finished the documentation.