**Socket Programming (Milestone)**

Group: 24

Members:

* Bryan Matthew (bmatth12)
* Christopher Stone (castone7)

1. **Message format:**
2. Register: register @handle\_A IP\_A P1 P2 P3

(**Notes:** Must not add extra space after the command, it’s counting for the number of words in the command. 1 handle, 1 IP, and 3 ports, don’t add “ “ after the ports)

* This command is used for registering a user to the tweeter application. It will take a handle (username), an IP address, and 3 ports (one for communication between the tracker and itself, and the two others for following)

**Example**: register @bryan 192.168.1.1 13000 13001 13002

1. Query handles: query handles

* This command will return to the client a list of handles that are currently registered in the tweeter application.

1. Follow: follow @handle\_A @handle\_B

* This command is used for controlling the “following” functionality. The order is: @handle\_A follows @handle\_B.

**Example**: follow @bryan @chris

1. Unfollow: drop @handle\_A @handle\_B

* This command is used for controlling the “unfollowing” functionality. The order is: @handle\_A unfollows @handle\_B.

**Example**: drop @bryan @chris

1. Tweet: tweet @handle message

* This command is used for sending a tweet (message).

**Example:** tweet @handle\_A hello world

1. End Tweet: end tweet

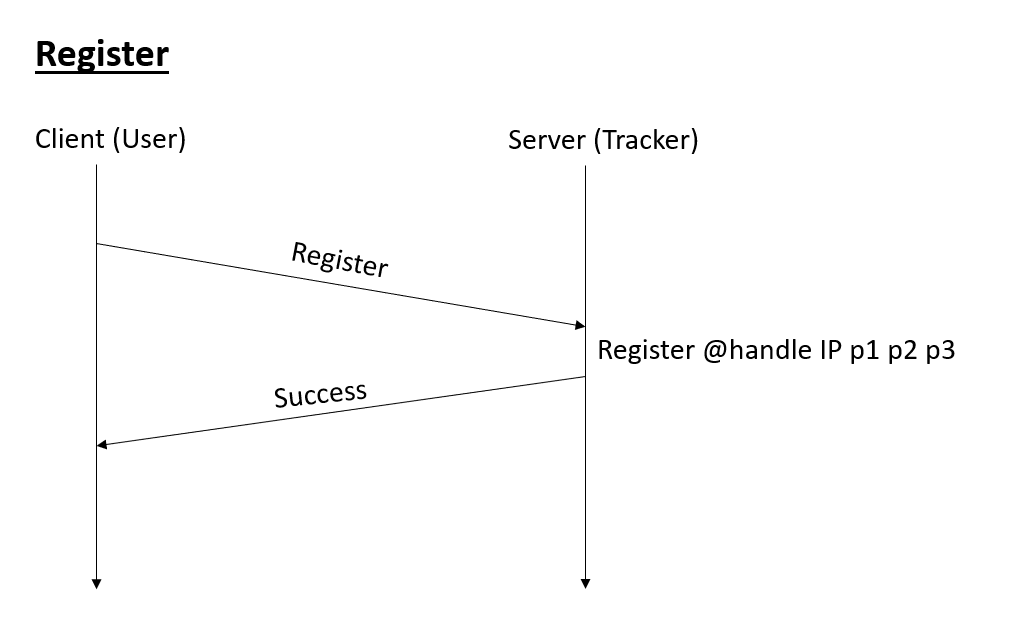
* This command is used for making sure all the tweets are sent in the correct order.

**Example:** end tweet

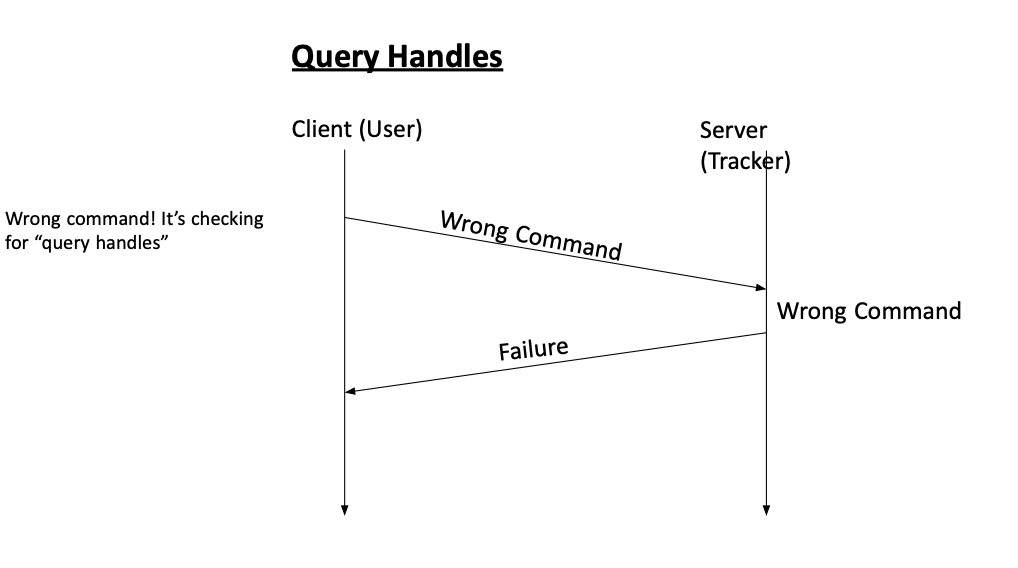
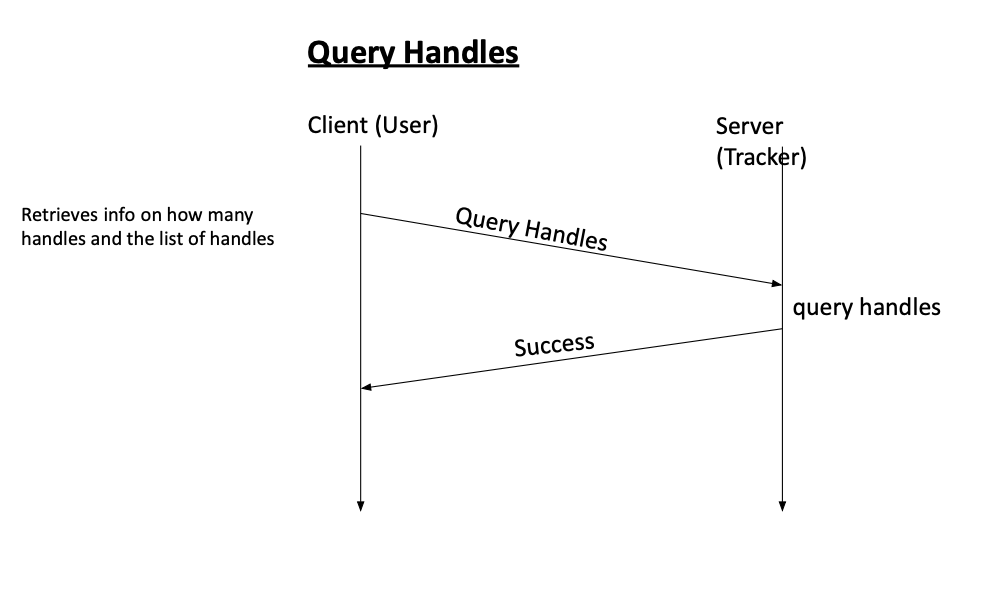
1. Exit: exit
   * This command is used for exiting the program.

**Example:** exit

1. **Time-Space Diagram**



**Diagram

Description automatically generated**

Diagram

Description automatically generated

1. **Data Structures & Algorithms Used**

- Queue is used to hold the messages, so it’s processed in the right (First-In-First-Out) order.

- List (array) is used to maintain the user table as it’s easy to sort a list and print it directly to the screen. We were considering the use of Hash-Map (dictionary), but we figured we didn’t need to access by keys in this case. Sorting is easy in list, but its time complexity is O(n log n).

- Dictionary (hash-map) — {Key: @handle, Value: List of followers} is used to maintain the follower table, as we can easily search for specific user to get the list of followers of that user. Searching in dictionary is O(1), that’s why we used it. We were trying to use list at first because it was simpler to sort the data in alphabetical order like we did with on the user table.

1. **Github Commits**

Link:

<https://github.com/ChrisrunnerR/TWITTER-TWEETER>

Graphical user interface

Description automatically generated

1. **Video Demonstration**

Link:

<https://www.youtube.com/watch?v=LWryZv0QWpY&ab_channel=ChristopherRunningVlogs>

**(Timestamp is in the description of the video)**

Starting Server - [0:00](https://www.youtube.com/watch?v=LWryZv0QWpY&t=0s)

Register - [2:05](https://www.youtube.com/watch?v=LWryZv0QWpY&t=125s)

Follow - [3:57](https://www.youtube.com/watch?v=LWryZv0QWpY&t=237s)

Query Handles - 5:45

Tweet - [6:00](https://www.youtube.com/watch?v=LWryZv0QWpY&t=360s)

End-Tweet [6:47](https://www.youtube.com/watch?v=LWryZv0QWpY&t=407s)

Drop - [7:50](https://www.youtube.com/watch?v=LWryZv0QWpY&t=470s)

Exit - [9:18](https://www.youtube.com/watch?v=LWryZv0QWpY&t=558s)