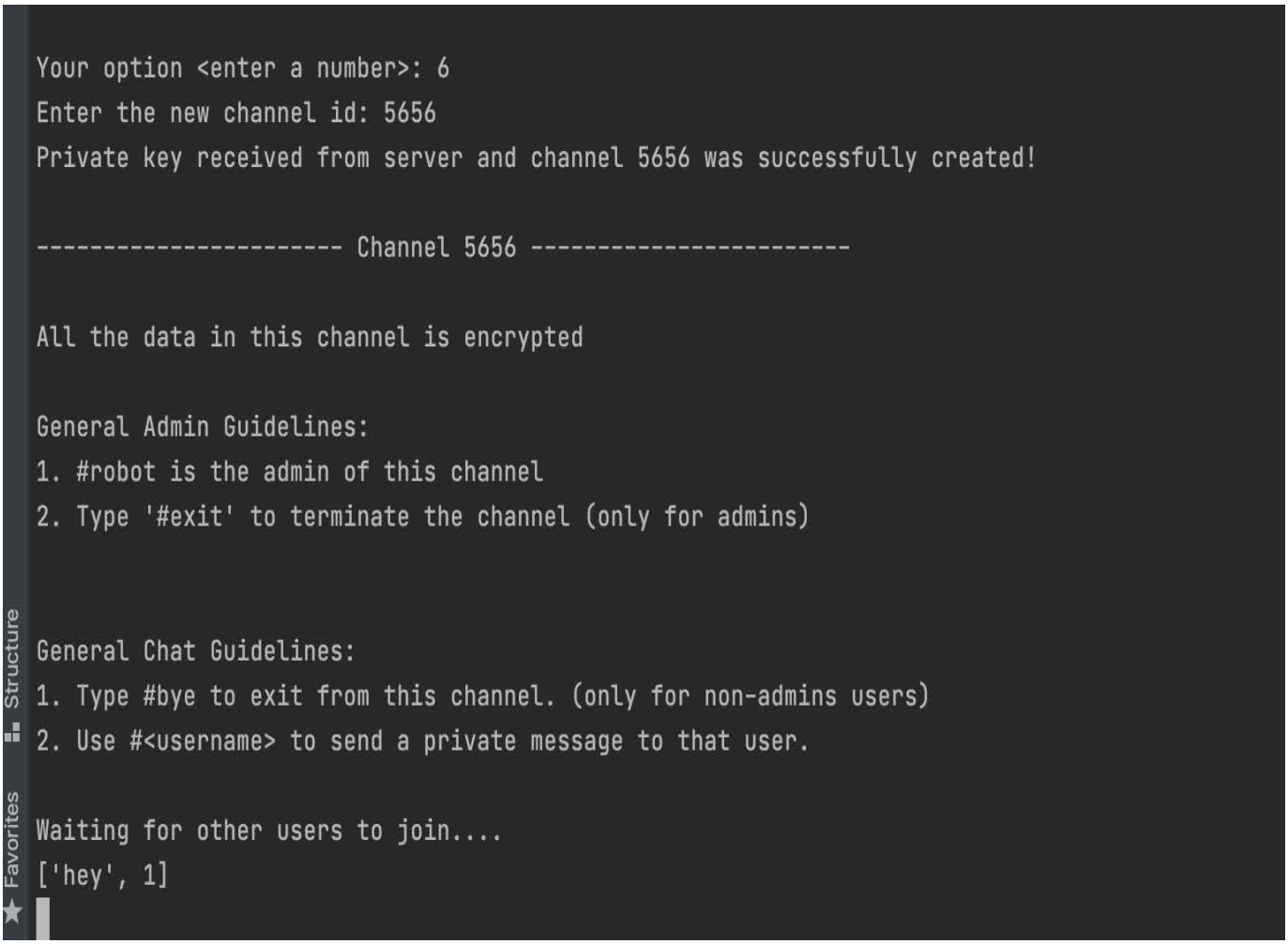
Please use this README file (or a new one) to provide the following documentation for this project:

* Your name and student id-
 - **Rupak Khatri, 920605878**
* General description of the project (a few sentences).
    - **Program to send the get the list of user and send and receive the message. Moreover we also use UDP protocal to send the message and use CDMA protocal to broadcast the message.**

* If you used external Python modules/libraries. Provide a requirements.txt file in this repository
  No external libraries other than discussed.
* Python version and compatibility issues (if any). Your project must be run exactly as in the running instructions
  described below in this file
  python above 2

* Attach screenshots or videos to this file to illustrate how your program works for all the options in the menu. This
is your proof that the project was working on your side.**Server side**
    -
    -



```
10.  Get the Routing Table of this client with Link State Protocol
11. Get the Routing Table of this network with Distance Vector Protocol
12. Turn web proxy server on (extra-credit)
13. Disconnect from server

Your option <enter a number>: 12

****** TCP/UDP Network ******
-----------------------------------
Options Available:
1.  Get users list
2.  Send a message
3.  Get my messages
4.  Send a direct message with UDP protocol
5.  Broadcast a message with CDMA protocol
6.  Create a secure channel to chat with your friends using PGP protocol
7.  Join an existing channel
8.  Create a Bot to manage a future channel
9.  Map the network
10.  Get the Routing Table of this client with Link State Protocol
11. Get the Routing Table of this network with Distance Vector Protocol
12. Turn web proxy server on (extra-credit)
13. Disconnect from server
```

```
Your option <enter a number>: 11


Routing table requested! Waiting for response....

Network Map:

          rupak    rumpum    robot    roos
------  -------  --------  -------  ------
rupak         0         1        2       3
rumpum        1         0        4       5
robot         2         4        0       6
roos          3         5        6       0

Routing table for roos (id: 55371) computed with Distance Vector Protocol:

          rupak    rumpum    robot    roos
------  -------  --------  -------  ------
rupak         0         1        2       3
rumpum        1         0        4       5
robot         2         4        0       6
roos          3         5        6       0
```

```
Your option <enter a number>: 7
Enter channel id you'd like to join: 5656
----------------------- Channel 5656 ------------------------

All the data in this channel is encrypted

rumpum just joined
robot is the admin of this channel

General Chat Guidelines:
1. Type #bye to exit from this channel. (only for non-admins users)
2. Use #<username> to send a private message to that user.

#<robot>hey
hey
```

```
Your option <enter a number>: 10


Routing table requested! Waiting for response....

Network Map:

          rupak    rumpum    robot    roos
------  -------  --------  -------  ------
rupak         0         1        2       3
rumpum        1         0        4       5
robot         2         4        0       6
roos          3         5        6       0

Routing table for roos (id: 55371) computed with Link State Protocol:

destination    Path                    Cost
-------------  --------------------  ------
rupak          (rupak,rumpum,robot,)      14
rumpum         (rupak,rumpum,)             8
robot          (rupak,)                    3
```

```
Your option <enter a number>: 9


Routing table requested! Waiting for response....

Network Map:

          rupak    rumpum    robot    roos
------  -------  --------  -------  ------
rupak         0         1        2       3
rumpum        1         0        4       5
robot         2         4        0       6
roos          3         5        6       0
```

```
our option <enter a number>: 13
venv) Rupaks-MacBook-Pro:client rupakkhatri$
```

3. *** A few sentences about all the challenges you found during the implementation of this project and how you overcame them. Please be honest here.**

For options I created method for them and added those methods to a list and when use requests them we will be calling them as the in1. for sending the menu class I first read the menu.py file as a string and then send it and change it to actual python format using exec method.

2. For connecting between classes and objects I created public variables that works outside the objectdex.

For the connection I used a class for binding, listening and accepting clients

For sending and receving data's I used a dictionary that make the work more efficent

for seeing unread messages. I added a separate method that have its own thread and works actively and store the result in unread message list