CS510: CODE REVISION AND REVIEW

Name: Deepa Hegde

PSU ID: 954680127

Email: hegde@pdx.edu

Table of Contents

CS510: CODE REVISION AND REVIEW ***1***

Project Details: **1**

About The Project: **1**

Proposed Changes: **2**

Progress on the Project: **2**

Challenges Faced: **2**

Future Scope: **3**

References: **3**

Project Details:

Name: Internet Relay Chat

Programming language: Python

GitHub link before refactoring: https://github.com/dshegde/Network-Programming/tree/70b1e2466f15bd127695e4f58f8c0636a1727962

GitHub link after refactoring: https://github.com/dshegde/Network-Programming.git

About The Project:

- This is my final graduate project for course CS 594 Internetworking Protocol in winter 2022 term.
- I had 2 other teammates working on this project along with me and I have the permission from them to modify this project.
- Internet Relay Chat (IRC) is a chat system on the Internet.
- It is a text-based chat (instant messaging) system.

- IRC is generally designed for group communication in discussion forums, called channels, but also allows one-on-one communication via private messages as well as chat and data transfer, including file sharing

Proposed Changes:

- Code refactoring (modularize, appropriate variable/function names)
- Adding docstrings/appropriate comments to the code
- Changing the UX with better/easy to understand instructions
- Error handling
- Update the README file to add steps to run the project.

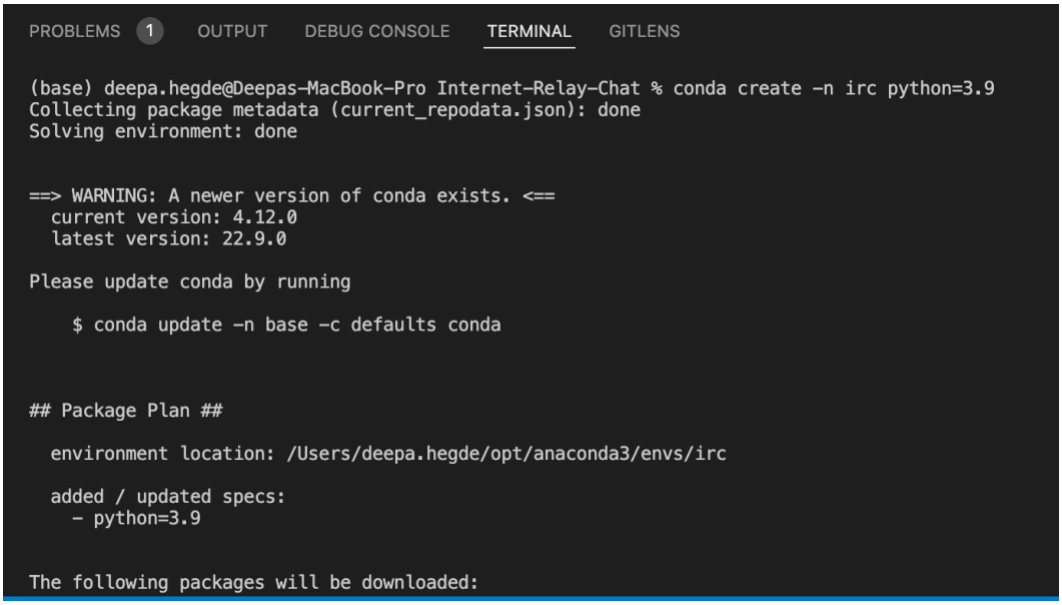
Progress on the Project:

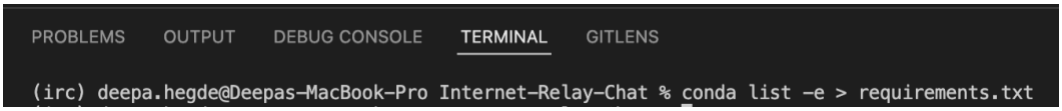
- Changed the project to follow the python standard PEP 8 style guide
- Added .gitignore file
- Config file to hold all the network connection details
- Connection class to abstract all connection related functionalities
- Code cleanup to remove redundant pieces of code
- Changed variable names to give a broader insight of what is it and its functionality.
- Appropriate method names that are self-explanatory
- Better choice of using inbuilt methods from socket module for socket/port connections
- Added functionalities for edge cases
- Added docstrings wherever necessary and added a single line comment for special functionalities
- Updated README.md file to contain all about the project from downloading and installations to running the application. Also added screenshots for better understandability
- Overall created 3 new python files, 1 new class, 1 .INI file, requirements.txt for the IRC (Internet Relay Chat) and .gitignore and README.md with respect to Git files
- Created a requirements.txt file for easy installations of the application across different environment

Challenges Faced:

- I tried handling the termination at the server side to capture keyboard interrupts using signal module and gracefully terminating with a user message. This did not work as there are multiple threads that are running, and the signal capture can only be used for a single/main thread. This changed did not allow more than one client to connect. This requires more reading on signal documentation to understand its implementation for multithreaded programs

- As I worked on this project again after almost a year, reading through the project with the variable names made it very hard to recollect the flow of the program. As a part of this project, I have changed the variable names for easier and quick understanding
- Writing the test cases for a multi-threaded program which involves client server communication. Below mentioned are the screenshots for how I generated the requirements.txt file





Future Scope:

- This is purely for feature enhancements
- Bug fixes along with more error handling
- Unit tests

References:

1. https://www.youtube.com/watch?v=-UwZQfOp64M
2. https://www.neuralnine.com/tcp-chat-in-python/