Devin Racaniello

Juan Arias

Software development I

7 May 2017

Abstract

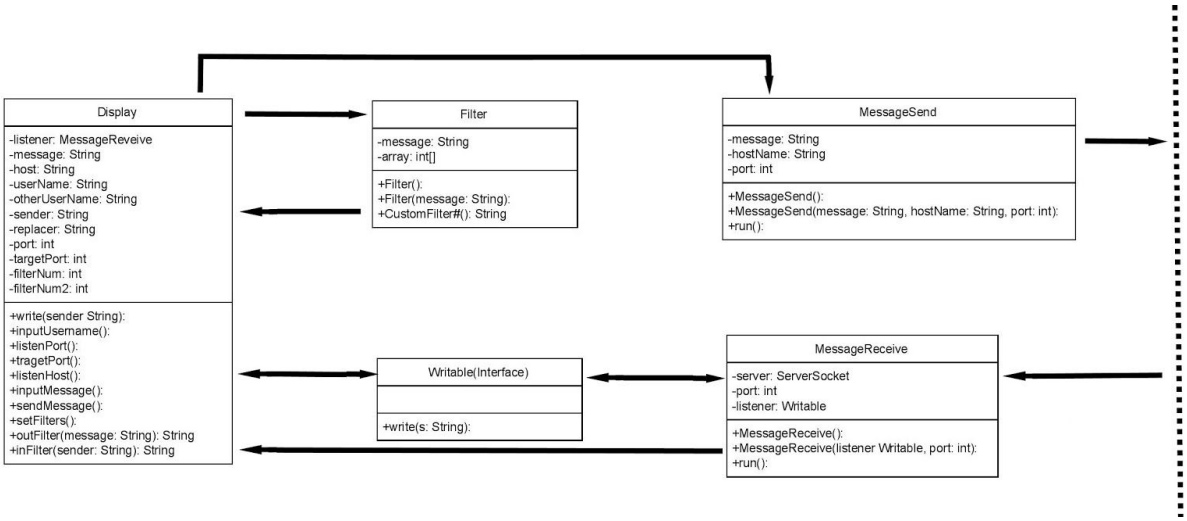
This product is a modified instant messenger. This program will have the capability to modify messages by the use of a filter system that filters outgoing and incoming messages. These features are realised through the use of java sockets and threading.

Introduction

This project is made to fulfill the purposes of communication, entertainment, customizability and security. By enabling the user to control the inputs and output of the messenger with filters they will have free reign over what they may use this program for. The project was designed with security in mind so those who are technologically savvy and aware of cyber risks as well as others who don't mind will be able to freely use this program.

This paper will include a description of how the program works, a description of how to interact with the program as well as a description of class interaction with a uml diagram. A requirements list will be provided. The paper will include a literature survey to compare the work to other products. There will be a user manual to describe how to use the product. Finally the paper will provide a conclusion and references to any materials.

Detailed System Description



The Systems mechanics have changed since the milestones submission of the system description. While initially I had attempted to make two separate programs one to host and one to connect to the host I changed the program to a more intuitive peer to peer design in which that is now unnecessary. The current system works by selecting an ip that both devices have access to and then both of the programs selecting a separate port and sending the information to the port that the other device is working on.

The program breaks down into six separate classes: MessageSend, Message, Receive, Filter, Display, Writable and MainChat. Writable is an interface in which the method Write() is created in order to transfer a message from different Classes of the Program to each other. Filter is the class that stores all the methods for filtering the message. MessageRecieve is where the program gets the socket inputs from the other user and sends along those messages to our local program. This process is accomplished through the use of overriding Thread's run method. MessageSend takes a string provided to it and sends it along to the desired host and port. The most integrated Class is Display. Display utilizes all the prior classes to make methods for the functions of the interaction and integration of the program. It holds all the methods for

getting information such as IP, port, username, and the message. It also stores information such as what filters are currently being used and with what specifications. The final Class is MainChat. MainChat is the file that runs Display methods and performs executions of the program. (Class Diagram is also in separate file).

Requirements

- Connectivity
    - Internet connections on both devices.
    - An IP that both users can connect to.
    - The knowledge of the port the other user is using.
- Program
    - A working program that connects to the required areas.
    - The ability to take input.
    - The Ability to send messages.
    - The ability to receive messages.
    - The ability to display received messages.
    - A message filter class.
- Users
    - Two people able to understand the program and use it.

Literature Survey

Whatsapp is an app that addresses very similar issues to those that my program focuses on. What's App is a messenger that uses the platforms of IOS, Android, Mac, Windows and Windows phone. It provides encrypted messaging. "Privacy and security is in our DNA, which is why we have end-to-end encryption in the latest versions of our app."(WhatsApp.com). Because

of the encrypted messaging this app provides communication and security. It is an app widely used amongst tech savvy individuals who value privacy. There are also many other messengers although Whatsapp seems to share a userbase with whom i would expect to benefit from my product. The biggest difference is their lack of customizable filters. They stick plainly to an end to end encryption base.

User Manual

Setting up

- To used this program you must be able to execute a java file in a way that allows for console display and inputs.
- Input your username as prompted by the program.
- Input a number for the desired filter.
    - Filter 1 replaces all character [space] with character -.
    - Filter 2 turns messages into ascii values.
    - Filter 3 encrypts the message.
    - Filter 4 decrypts encrypted messages.
    - Filter 5 prompts you to enter a replacer, the replacer will replace the first letter of every word of a message with the replacer
    - Filter 6 will reverse the string of a message
- Input the hosting IP you will connect to.
- Input the port of the other user you would like to connect to.
- Input the port you would like to run on that the other user will connect to.

Use Once set up

- Type in any message and hit enter to have it sent.
- Type /setfilter and hit enter to change filters.

    - The Filter prompt from the beginning will reappear and you fill it according to the filters described earlier.
- Type /setusername and hit enter to change your displayed name.
    - Simply enter in a username and hit enter.
- To end the program just exit out of the messenger.

If a user desires the use of custom filters there is no official way within the program to create such things but if one edits the Filter class and adds their own filter and updates the Display class to allow it to be recognized by the prompt then a filter can be added. The program will still be able to communicate with an unedited program.

Conclusion

This program is a java based messenger with incoming and outgoing filters. It provides users with communication, entertainment, customizability, and security. The filters will allow people to either send fun, usefully modified, or encrypted messages. There are similar programs in the messengers provided by other companies although none are completely like this program due to the unique filtering system. The program works through a peer to peer socket connection system that utilizes threads. It is used by running the program establishing a connection and settings and then simply sending messages to the other.

References

"WhatsApp FAQ - End-to-End Encryption." WhatsApp.com. WhatsApp Inc., n.d. Web. 07 Apr. 2017.

Liang, Y. Daniel. *Introduction to JAVA Programming: Comprehensive Version*. 10th ed. Upper Saddle River, NJ: Pearson/Prentice Hall, 2015.