We created a messaging application with a Graphical User Interface using Python 3.x. Our code uses the Python tkinter GUI module to layout buttons, form fields, and a scrollable text box for viewing messages. Messages are stored in a file whose name matches the username of the recipient.

Our application allows anyone with access to the Trace directory to run the program and send/view messages. A username is not required to use the application, and in fact, if no username is entered, the message is marked as “Anonymous”. There is no password system in place, so anyone can pose as anyone else and send or read messages. Upon entering a username and clicking LOAD MESSAGES, the user’s messages are loaded into a text box which automatically scrolls down to the bottom of the window so it always displays the newest messages.

We added code which periodically reloads the user’s file to retrieve and display new messages without manually reloading the messages using the GUI button. This allows multiple users to communicate with each other in a semi-live environment, much the same way that an instant messaging program would work.

With the current version of our application, we are using hard disk space for message storage, memory (RAM) for the operation of the Graphical User Interface, and the CPU to process the instructions.. To use the service across the internet, we would need to add an interface to communicate over the TCP/IP protocol. Additionally, we would need to set up what is currently the Trace folder as a public-facing server. Of course to do that, we would need to implement a strong username and password system which would ideally use a person’s email address as their identity and would verify the email address before allowing access to the system. It would be necessary to encrypt all messages and communications to prevent others from eavesdropping on conversations.

We currently do not have a limitation imposed on file size, so as the number of users increase, the overall disk space needed will probably get out of control. We would need to impose some sort of limitation on the number of messages retained, either by specifying some maximum number of messages (such as the last 1,000 messages) or some maximum time-frame (such as the last 90 days).