

# Term Project: JavaMess

## Post Delivery Write-up

### **Introduction:**

This document is short analysis of the JavaMess program. This document will discuss, the use and features, cut features with an explanation, and possible future improvements.

### **Scope and purpose:**

This document is to help familiarize instructors and possible students reading this document of program, past problems, and future considerations for refinement and polish.

### **Use of JavaMess:**

The JavaMess chat system is a java based real time chat communication tool that used multithreaded programing and JavaFX elements for a GUI. The end user uses the client to interface with the program by connecting to the server. In order for a user must register with a username that does not contain the character 'X' or one that has been previously registered. Once a user has registered, they may login with their username and password. Users are then able to send public messages to all online users with the send button. Private messages can be sent by selecting a user and hitting the private message button once a message has been entered. The previous history of private messages can be retrieved by selecting a user and pressing the get history button.

## **Cut Features:**

When this program was originally designed there were some additional features that were there to help make it more feasible practical use outside the classroom. Some major features ended up being cut largely due to inexperience and time costs associated with learning how to implement the feature. The most glaring of missing features is the lack of security. Originally I intended to salt and encrypt passwords before storing them in a file. Currently the security is woefully inadequate since the passwords are stored in plain text. There were also designs for separate windows for each private conversation. Catching private messages from another user became a concern and may be addressed if future versions are developed.

## **Possible Future Improvements:**

JavaMess, while a completely functional chat system, has great potential to grow. There are some constraints that were introduced as a temporary proof of concept that could be developed further in order to allow for it to be scaled up. For example, the number of threads available on the server currently is set to 10 by an array, which is severely limiting. The number of user chat files that could be generated by this system could quickly grow to be unmanageable unless there is a system put into place to organize chat logs into directories. If this program were to be taken online, password security would need to be a high priority.