

# FOXCHAT

## Project Milestone



**Mark Blankson-Hemans**

# TABLE OF CONTENTS

<b>ABSTRACT .....</b>	<b>1</b>
<b>INTRODUCTION .....</b>	<b>1</b>
<b>DETAILED SYSTEM DESCRIPTION .....</b>	<b>2</b>
UML Diagram(s) .....	3
<b>REQUIREMENTS .....</b>	<b>4</b>
<b>LITERATURE SURVEY .....</b>	<b>4</b>
<b>USER MANUAL .....</b>	<b>5</b>
<b>WORKS CITED .....</b>	<b>6</b>

# ABSTRACT

Foxchat is a Java program developed to be used by Marist students to send messages between one another within the Marist campus. Using java socket programming and the java swing class, users are able to use the lines of code from the 'back-end' through a GUI over a server-client connection to send texts between one another. The program was developed to save costs from text-messaging, and to also securely connect students on the Marist campus in a matter of seconds using only their unique CWID's assigned to them.

# INTRODUCTION

Foxchat was developed in order to explore the benefits of Java socket programming. The program utilizes the more expansive and user-friendly adaptation of java sockets: **stream communication**. A stream connection is also known as implementing a Transfer Control Protocol (TCP). This was developed to meet the goal of the project – a server class and a client class.

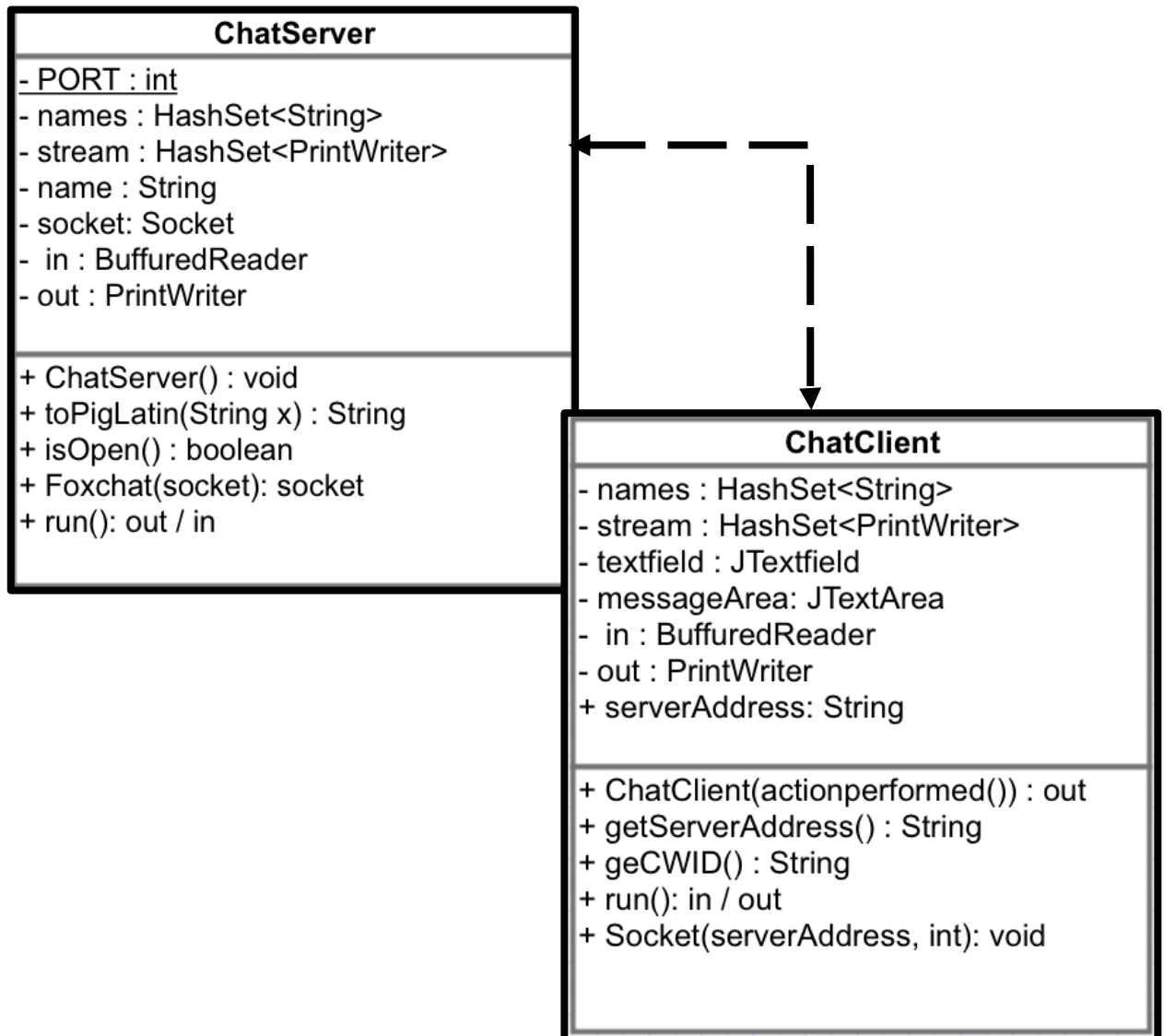
# DETAILED SYSTEM DESCRIPTION

Foxchat is comprised of two distinct classes, the server class to establish a connection and listen for contact, and a client class to request connection and send data to the server for it to be processed.

Foxchat is possible due to the existence of three core packages that are available in the most current java version (Version 8 Update 131). These three packages are but are not limited to:

- *Java IO*
- *Java Net*
- *Java Swing*

## UML DIAGRAM(S)



# REQUIREMENTS

Objectives	Successes
The system is supposed to facilitate the sending of messages between students at Marist College without the need of an internet connection.	The system is able to do this.
The ability to have chats between more than one user.	The system is able to do this.
The system is easy to use and is user friendly.	Achieved.

# LITERATURE SURVEY

Similar experiences with the `PrintWriter` function have sighted the use of the `flush()` method, however, due to inexperience and lack of time with the project, it was unable to be used to rollback discrepancies found when data was being validated.

In the `FoxChat` program, the current structure uses a try-catch structure to implement the `PrintWriter` functions, and the `printWriter` function was set up to automatically flush code

(committing it to the server) using the **true** java keyword when the predefined method was called.

# USER MANUAL

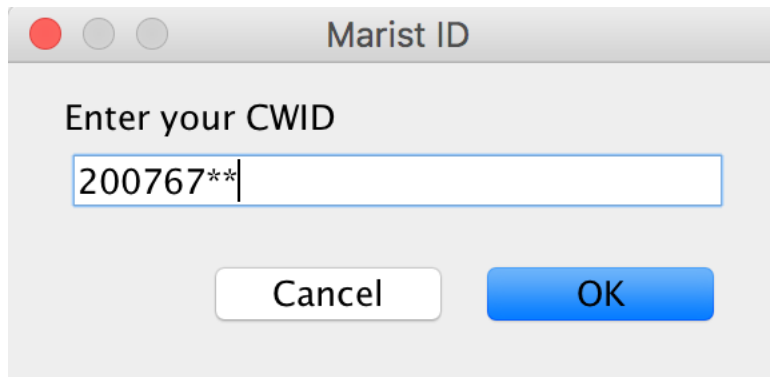
To run the program, the latest Java environment as of May 2017 needs to be installed on the computer. The computer should also have a Java Virtual Machine to process the interfaces as well as run the classes. The computer also needs to have network capabilities, both to run the server class as well as the client class.

For the current extent of this program, a computer shall be dedicated as the “server” for the application, although future developments may prove otherwise. As such, the ChatServer class should be run before any clients connect.

Running the Client program should create the first interface which confirms that the server has been successfully created with a network port predetermined.



Proper input should lead to the next prompt asking for a CWID number. This would also serve as a client’s “Username”.



Correct input should lead to the next interface for users to begin chatting with others. The textbox at the top of the pane is used to collect input, whereas the textarea below displays the messages. Messages are transmitted through the commitment of a line, which is done by pressing the enter key.

## WORKS CITED

Oracle, Java. "Class BufferedReader." *BufferedReader (Java Platform SE 8 )*. Java, 12 Jan. 2016. Web. 12 May 2017.

<http://www.cs.utexas.edu/users/mitra/csSpring2004/cs313/assgn/BufferedReader.html>

<https://docs.oracle.com/javase/7/docs/api/java/io/PrintWriter.html>

[http://www.tutorialspoint.com/java/io/printwriter\\_flush.htm](http://www.tutorialspoint.com/java/io/printwriter_flush.htm)

<https://www.tutorialspoint.com/java/io/index.htm>