



## Assignment Cover Letter

**(Individual Work)**

<b>Student Information:</b>	<b>Surname</b>	<b>Given Names</b>	<b>Student ID Number</b>
1.	Lumban Tobing	Zefanya	2201796970

<b>Course Code</b>	<b>: COMP6335</b>	<b>Course Name</b>	<b>: Programming Languages</b>
<b>Class</b>	<b>: L2AC</b>	<b>Name of Lecturer(s)</b>	<b>: 1. Minaldi Loeis 2. Jude Martinez</b>
<b>Major</b>	<b>: CS</b>		
<b>Title of Assignment (if any)</b>	<b>: Chat Application</b>		
<b>Type of Assignment</b>	<b>: Final Project</b>		
<b>Submission Pattern</b>			
<b>Due Date</b>	<b>: 2-9-2019</b>	<b>Submission Date</b>	<b>: 2-9-2019</b>

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer's instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

### Plagiarism/Cheating

BiNus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

### Declaration of Originality

By signing this assignment, I understand, accept and consent to BiNus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student:

1. Zefanya Gedalya B.L.T

(Name of Student)

## Table of Contents

I.	Project Specification .....	1
	1. Problem .....	1
	2. Solution .....	1
II.	Solution Design .....	2
III.	Discussion .....	6
	1. Implementation .....	6
	2. How it works .....	6
IV.	Evidence .....	8
V.	References .....	10

# Project Specification

## Problem

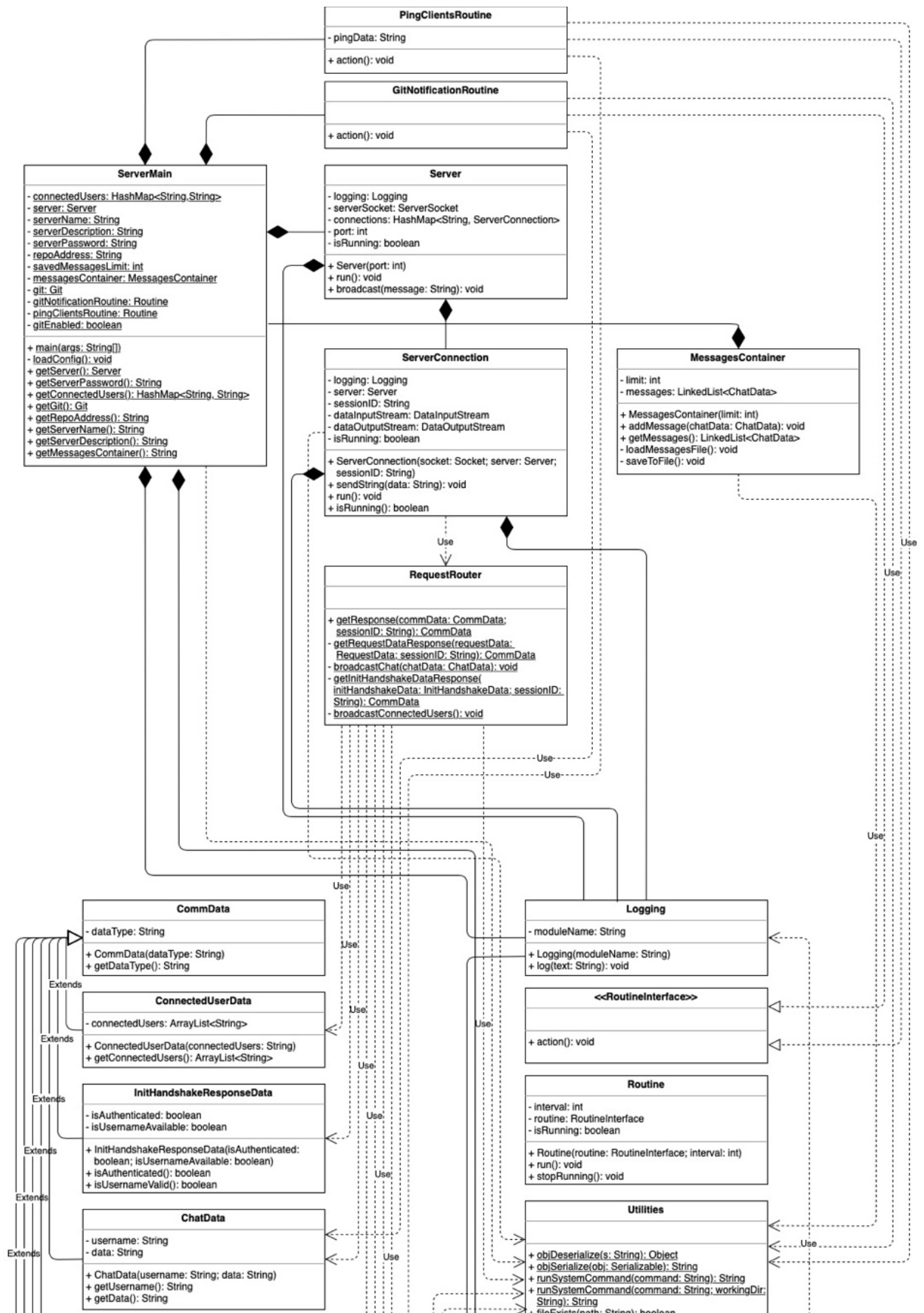
A lot of new CS students working on a group project faces the same problem. Communicating and working on files together remotely. Files are zipped, sent through social media platforms, modified on one end and not the other and everything is just a complete mess. Modifications are not synced, no one knows who does something, etc.

## Solution

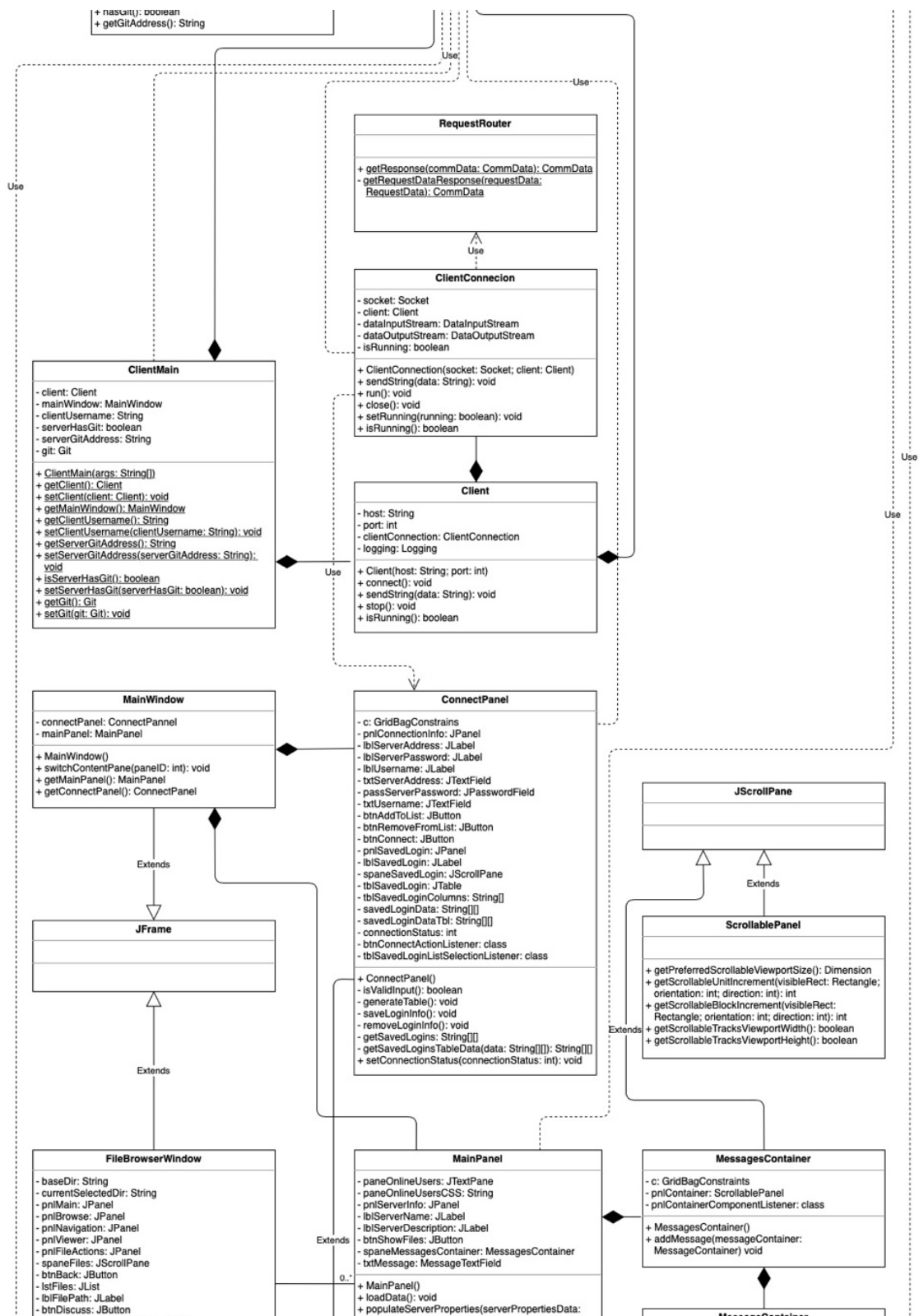
To solve the problem stated above, I am making a chat application that is teamwork focused. It will integrate with a git repository. The application should have features as follows:

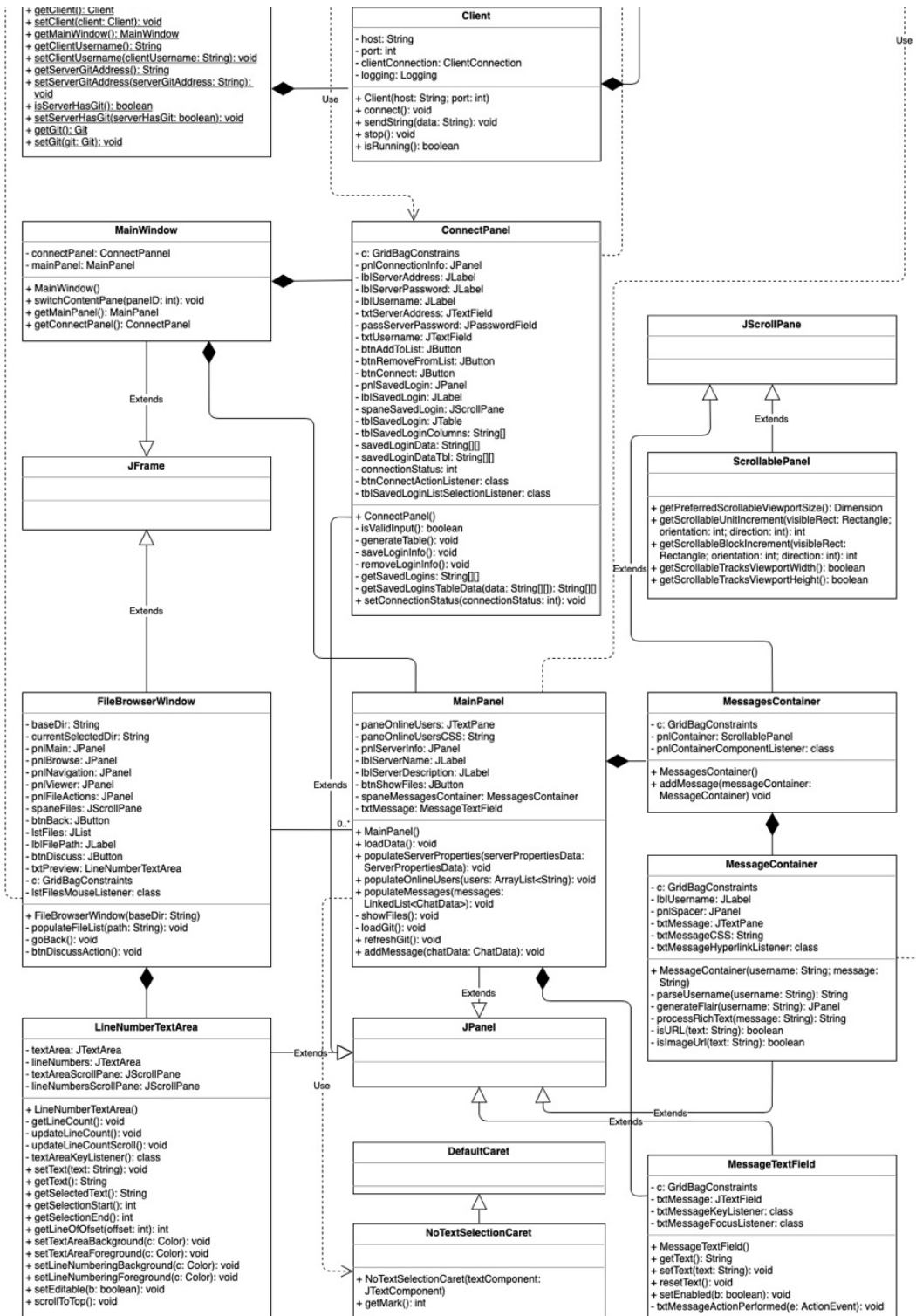
1. Self-hosted, configurable server
2. Password protected server
3. Git update notification
4. File viewer for the set repository
5. Code discussion feature (discuss selected lines of code in the file viewer) with detailed file path, line range being discussed, the contents of the code and the topic of the discussion.
6. Text styling (bold, italic, and strikethrough)

# Solution Design









# Discussion

## Implementation

This project is made possible with the variety of built-in libraries in Java. This project uses a lot of Java libraries. The most important libraries that made this project possible are AWT, Base64, Serializable, ServerSocket, Socket, Swing and Thread. Other than that, it requires git cli installed on the system.

AWT and Swing provides a simple and easy way to make a graphical user interface. In this project, it is used to make a graphical user interface for users to interact with. It has a lot of components like button, panel, text fields, etc. that is useful in the making of this project.

ServerSocket and Socket provides a way to setup a TCP server and client easily. In this project, it is used to send data in string from server to client and vice versa. This is essential for this project because it requires a client-server communication to work.

Serializable lets an instance of a class to be represented in a sequence of bytes or vice versa. This is very important as it enables an instance of a class to be stored somewhere else or sent through the network.

Base64 library provides a way to encode and decode bytes to and from base64 easy. In this project, it used to encode bytes of serialized instance of a class. The encoded bytes contain information that is sent through the network.

Thread library lets processes and operations to run at the same time. This library is very important for this project as a lot of actions has to be done concurrently. It is used on socket connections and routines that should be executed every n second/s.

Git command line interface provides a way to interface with a git repository inside the app. It is used for repository change notifications and code discussion.

## How it works

A connection is made between client and server with Java's built-in socket library. It communicates using TCP protocol. To make the server accessible from the outside of the local network, port forwarding is required or reverse shell to open the port at a remote address.

ServerSocket works by returning a socket instance when a connection is accepted. The accepting method of ServerSocket is blocking, meaning no other operations can be done until it is finished. This is where threading comes in. A ServerSocket is put in a class that extends thread with overridden load method waiting for an incoming connection. When a connection is accepted, the socket object returned by the accept method is stored in an instance of a connection class.

Connection class handles the process of receiving, processing and sending a message to a client. The connection class is a thread making concurrent actions on different clients possible. The instance of connection class is stored in a dictionary with a randomly generated UUID as the key and the instance of connection class as the value. This is made to identify



which client is connected to the server. UUID is generated every time a socket connection is accepted.

When a client disconnects, it will tell the server that it is disconnecting, and the server will stop and remove the connection class from the hash map. If a client disconnected unexpectedly (client closed forcibly, connection loss, etc.), the connection will be cleared automatically when the server checks for connection every 10 seconds. This made sure that no unnecessary system resource is used.

The data sent between the client and the server are bytes of serialized objects encoded in base64. This method is used because some data needs to be sent simultaneously. By creating a class for each data that is going to be sent containing specific information it made the process easy, straightforward and less error prone than parsing a string manually.

Client connects to user specified host address along with the server password and visible username. Server password is used to protect the server from unauthorized access. Username is used for the users to identify each other. Username must be unique and must only contain letters, numbers, underscore and period. Pressing the connect button will connect itself to the specified host address and authenticate. If a connection is successful and user is authenticated, the user will be presented with a chat window with a list of currently online users, messages history, server information, and a text field to enter a message. Users can type on a text field and press enter to send the message. Users can also press shift + enter to make a new line. Users can also add text styling like bold, italic and strikethrough with tags like [b], [i] and [s] with their corresponding closing tag like [/b], etc. This is possible because Java's Swing library has text panes that supports HTML and CSS enabling rich text styling.

If the server has a git repository linked, the client will show a button that when clicked will show a file explorer to view all of the files in the git repository. When you select a file, a preview of the file is shown on the right side of the file explorer. Users can select/highlight a part of a code or a regular text file and discuss the specific lines of code easily. It will show the file location, selected text, and the discussion message in the main window so everyone can look into it easily. It also has tags to differentiate a regular message and a code discussion.

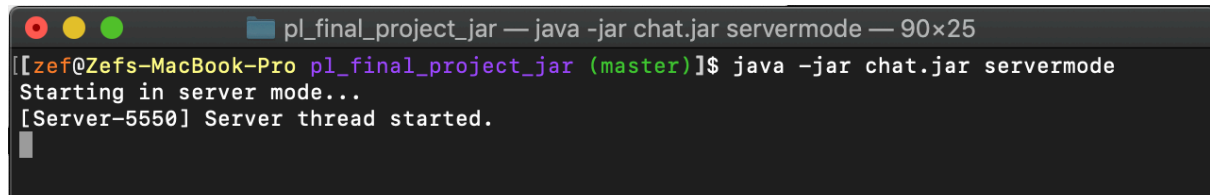
Any modification to a git repository will be visible via notifications. It works by checking the last commit on a repository and comparing it to the latest one. If there are changes, it will tell the clients that changes had been made to the repository and tell the client to pull all changes on the repository. Git works by interfacing with git cli program by executing git commands in the terminal. Git installation is detected by running "whereis git" command on the terminal. If it is installed, it will return the location of git. If it is not installed, it will return an empty string.

Checking git updates and disconnected users needed to be done regularly. A routine class will run specified actions wrapped in a class that interfaces RoutineInterface. Routine class runs on a thread. It runs the actions passed, sleeps the thread for n seconds, and do the actions until the thread is stopped.

Users can decide whether to host or to use the app with the same jar file. To launch in client mode, users simply can open the jar file regularly. To start in server mode, run "java -jar chat.jar servermode" in the terminal.

## Evidence

This section contains screenshots of working program evidence. The source code and the program file can be downloaded from <https://zefryuuko.github.io/pl-final-project/>



```
pl_final_project_jar — java -jar chat.jar servermode — 90×25
[[zef@Zefs-MacBook-Pro pl_final_project_jar (master)]$ java -jar chat.jar servermode
Starting in server mode...
[Server-5550] Server thread started.
```

Figure 1: Server running in server mode

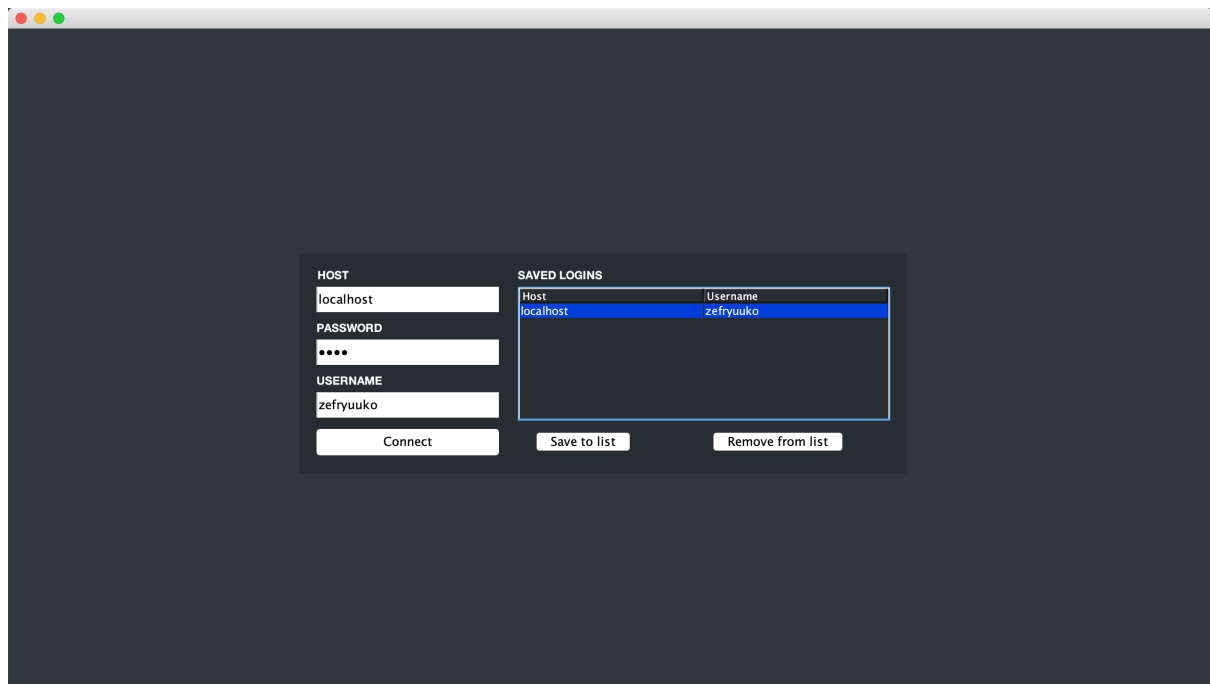


Figure 2: Client login screen

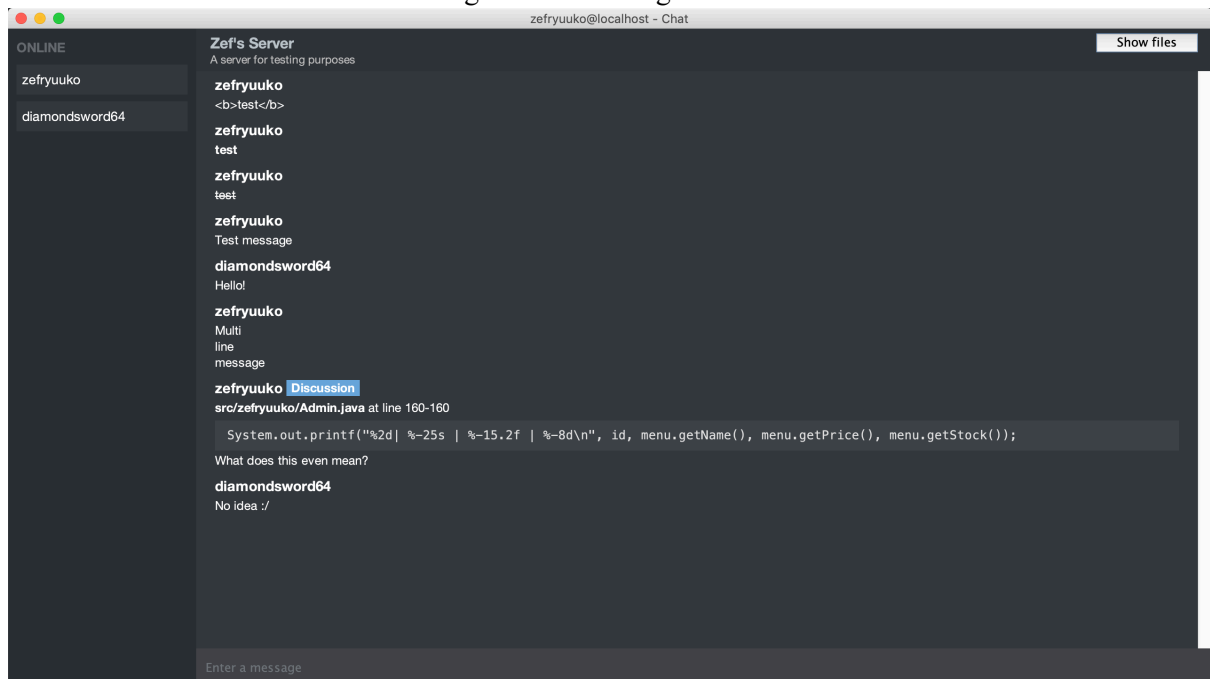


Figure 4: Messaging screen

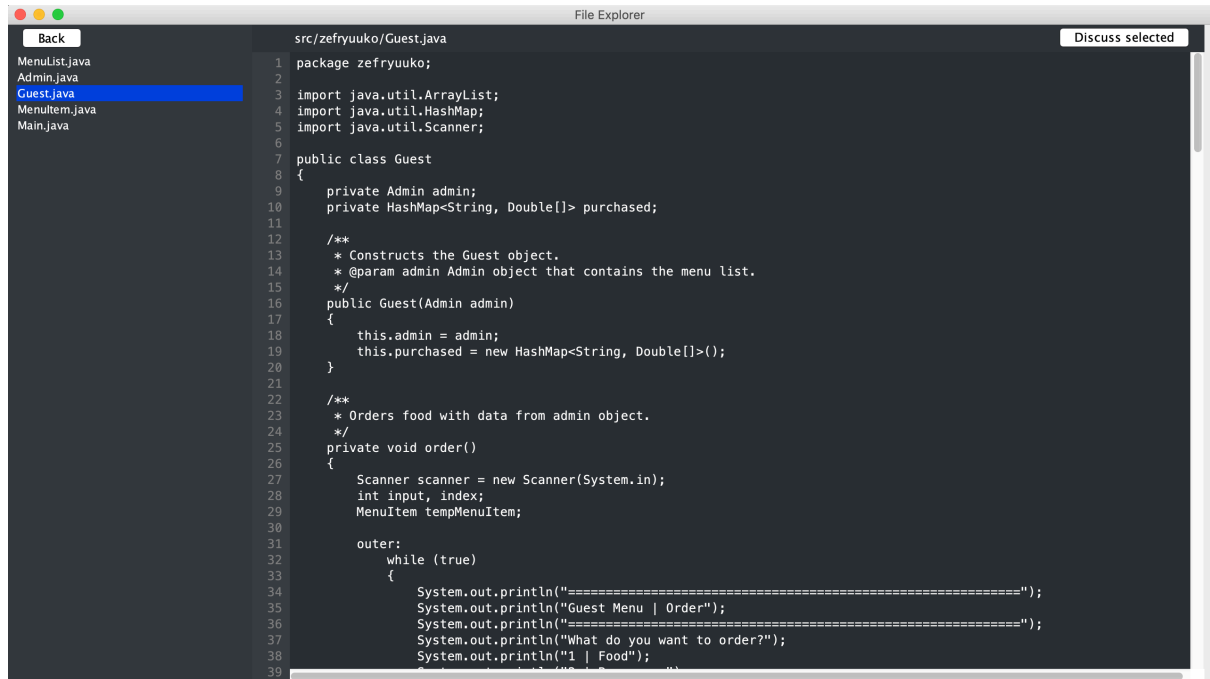


Figure 5: File explorer window

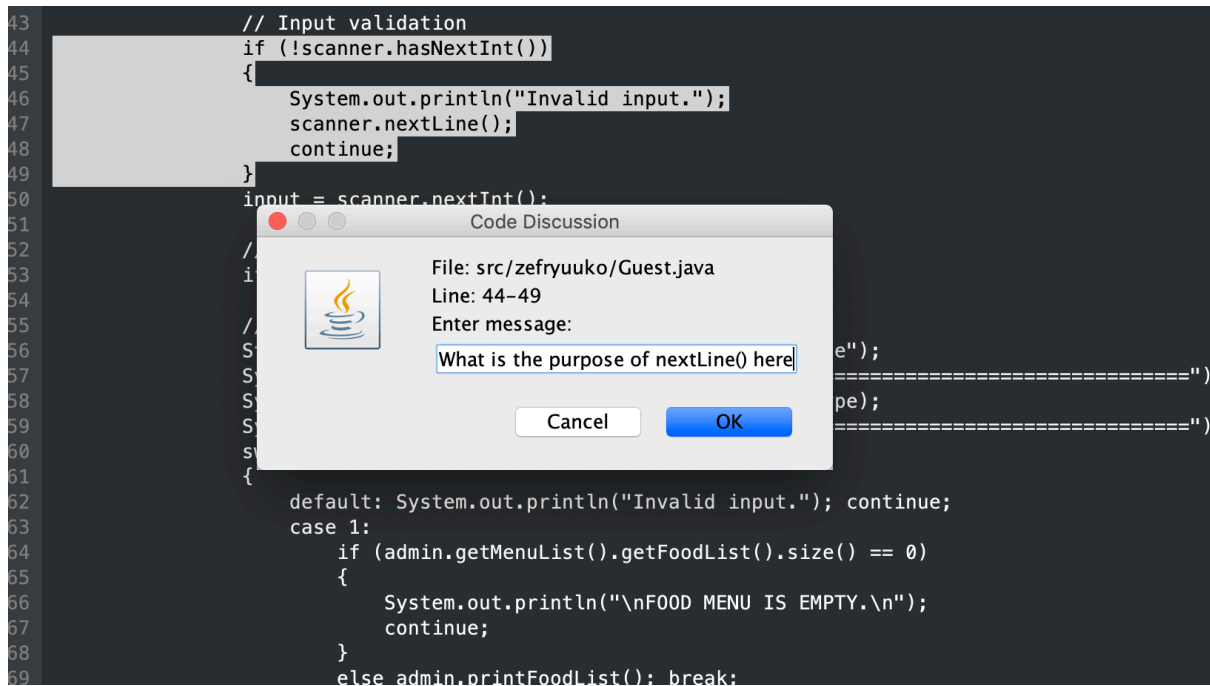


Figure 6: Code discussion feature

## References

- Java Networking: <https://www.youtube.com/watch?v=JV3YsR9wiq0&t=1215s>
- Object-String serialization: <https://stackoverflow.com/questions/134492/how-to-serialize-an-object-into-a-string>
- Running system commands in Java: <https://stackoverflow.com/questions/15356405/how-to-run-a-command-at-terminal-from-java-program>
- Delete directory with files in Java: <https://stackoverflow.com/questions/20281835/how-to-delete-a-folder-with-files-using-java>
- Operating System Detection: <https://www.mkyong.com/java/how-to-detect-os-in-java-systemgetpropertyosname/>
- JScrollPane with fixed width: <https://stackoverflow.com/questions/2716274/jscrollpane-needs-to-shrink-its-width>
- Disable selection on TextArea: <https://stackoverflow.com/questions/32515391/how-to-disable-text-selection-on-jtextarea-swing>