

Chat and file sharing app



**Introduction**

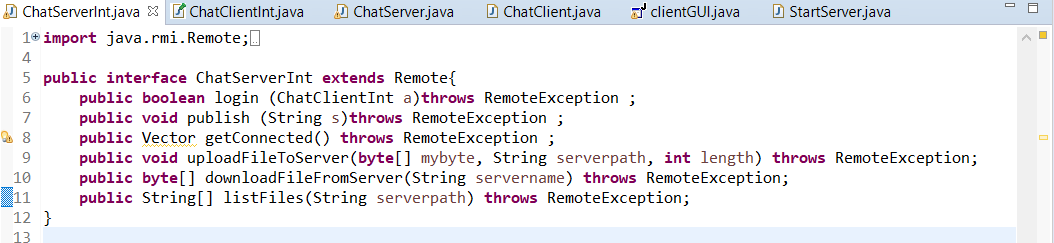
A Client-Server Application using RMI which implements functions like Chat between clients and file transfer. The objective is to create an application that allows multiple clients to connect to a main server and chat amongst themselves and communicate and also share files. It’s a simple solution providing clients to connect and interact with each other.

**Classes**

Server

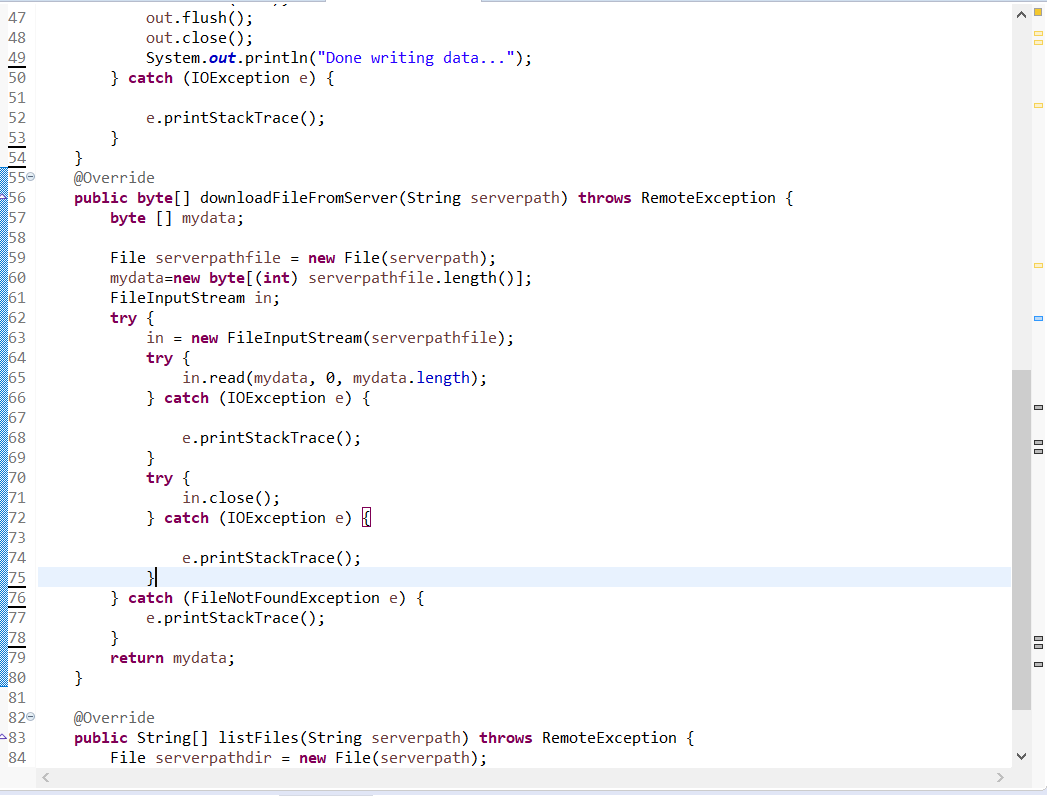
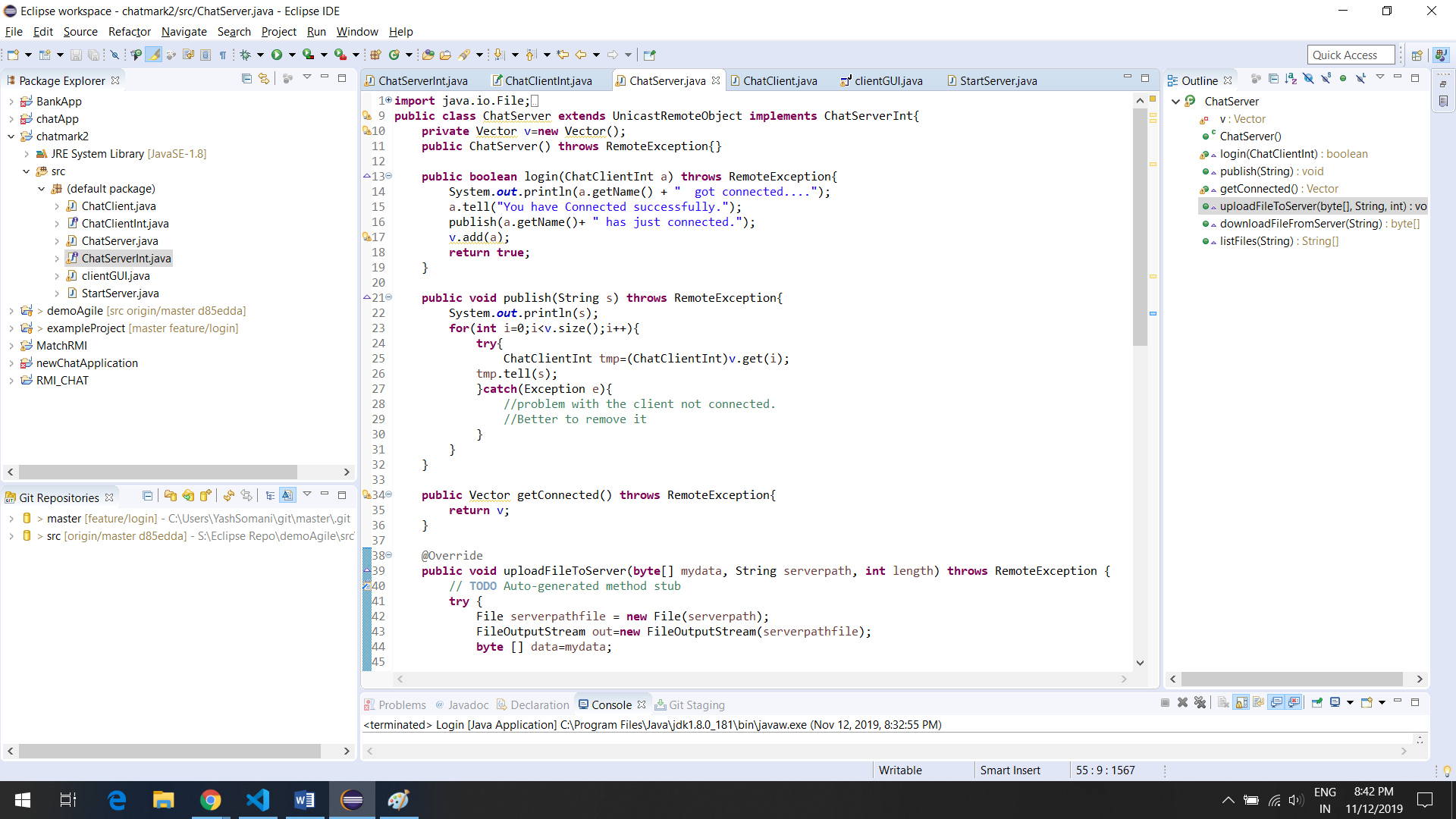
1. ChatServerInt

This class is the interface class for the server. It declares functions like



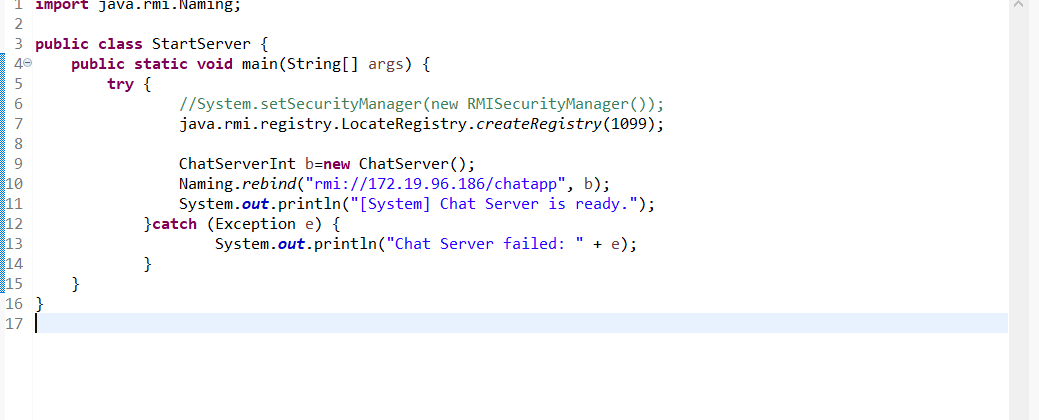
1. ChatServer

This class is the Implementation class for the server that contains all the functionality of the above methods.



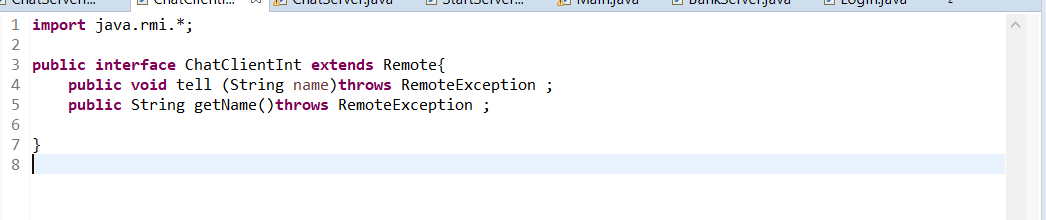
1. StartServer

This is the Server class that creates the Registry.



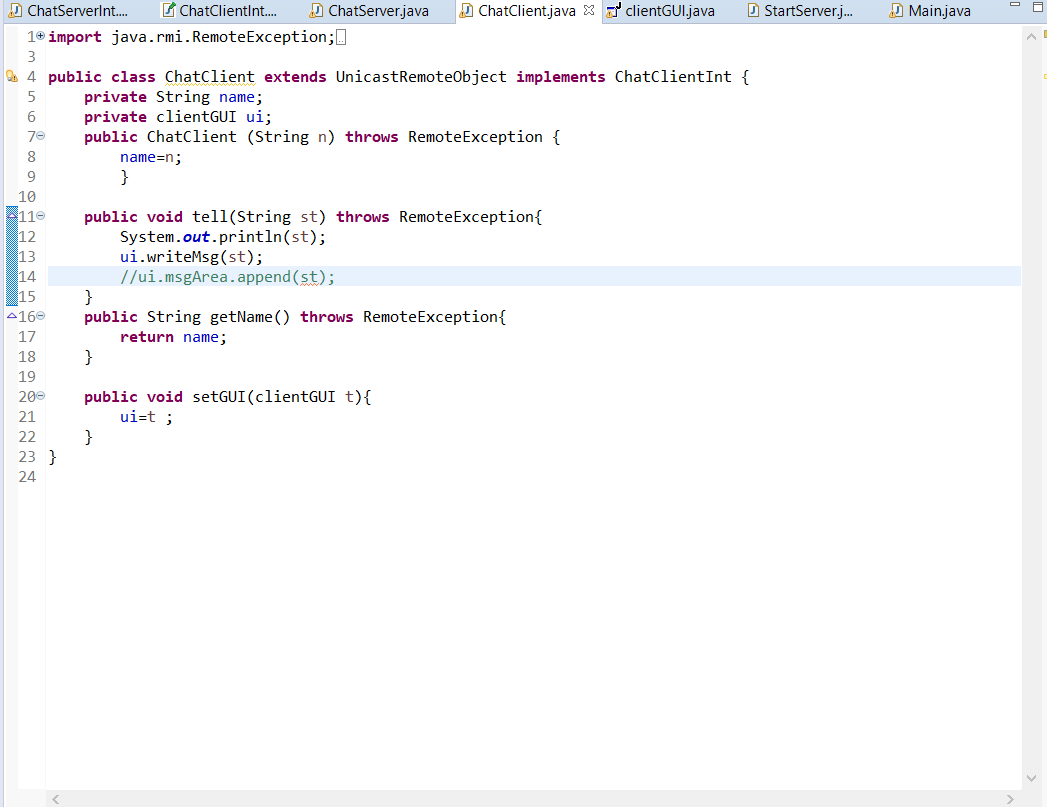
4)ChatClientInt

This is the client interface containing functions.



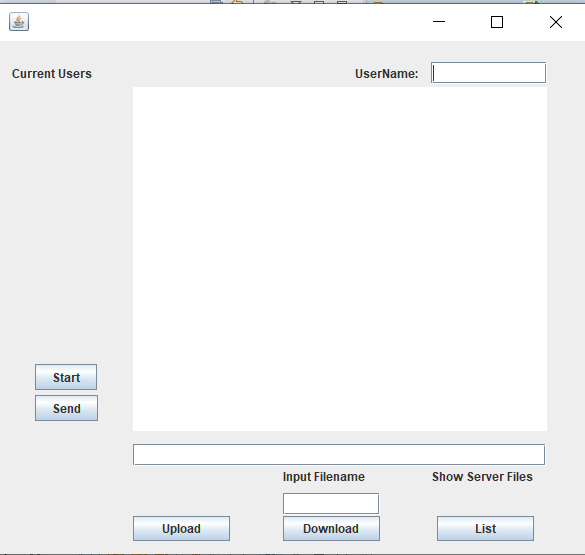
5)ChatClient

This is the implementation class of the above interface



6)clientGUI

This is the client class that connects to the server. GUI built using window builder



**Features** :

1. Normal Chat

It allows many clients to connect with the server and interact with each other. Basically a client specifies an IP address for the server and connect, after which communication is online and you can send your messages and others sending their messages.

1. All Online

It allows users to check who all are currently connected to the system. This is simply done by creating a Timer and calling a function updateUsers.

1. List all the server files

It allows users to check which all files are currently in the server directory and show them in a JList.

1. Upload files

It allows users to upload their own files using FileDialogBox to the server so that other clients can download it to their own system.

1. Download Files

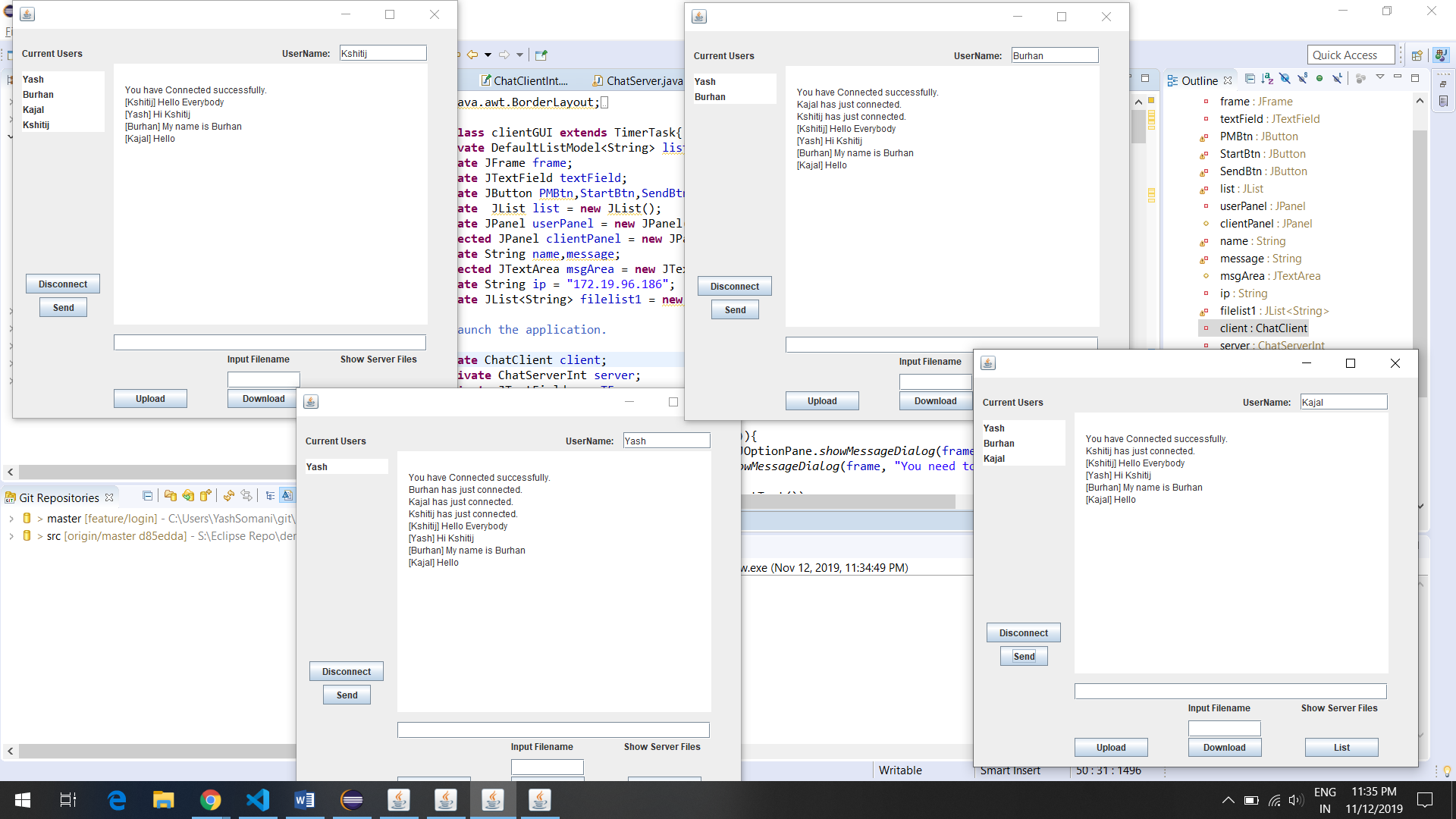
It allows users to specify which file they want by typing the correct filename in the textField and click download.

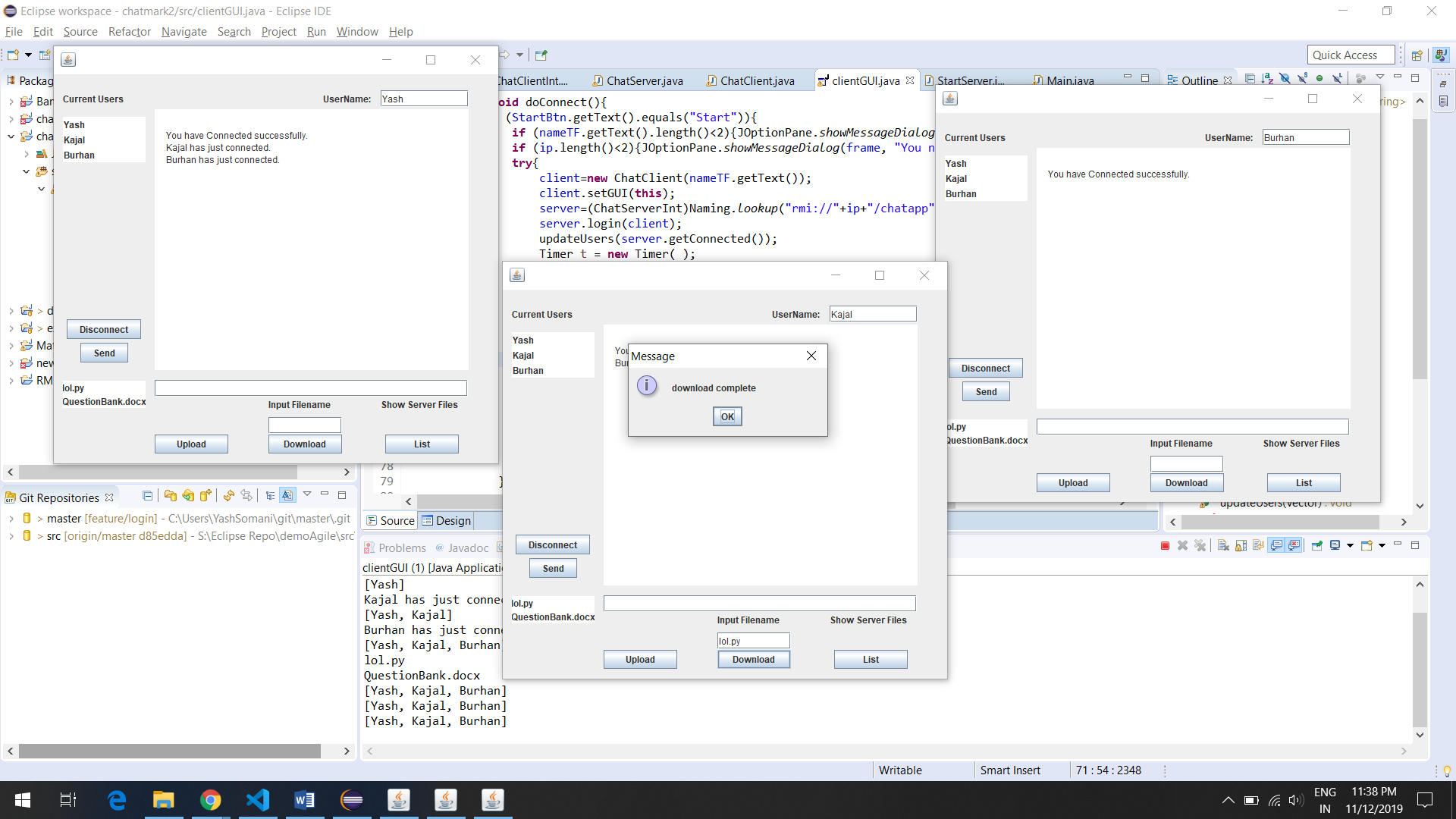
**Steps to follow while importing the project into your workspace:**

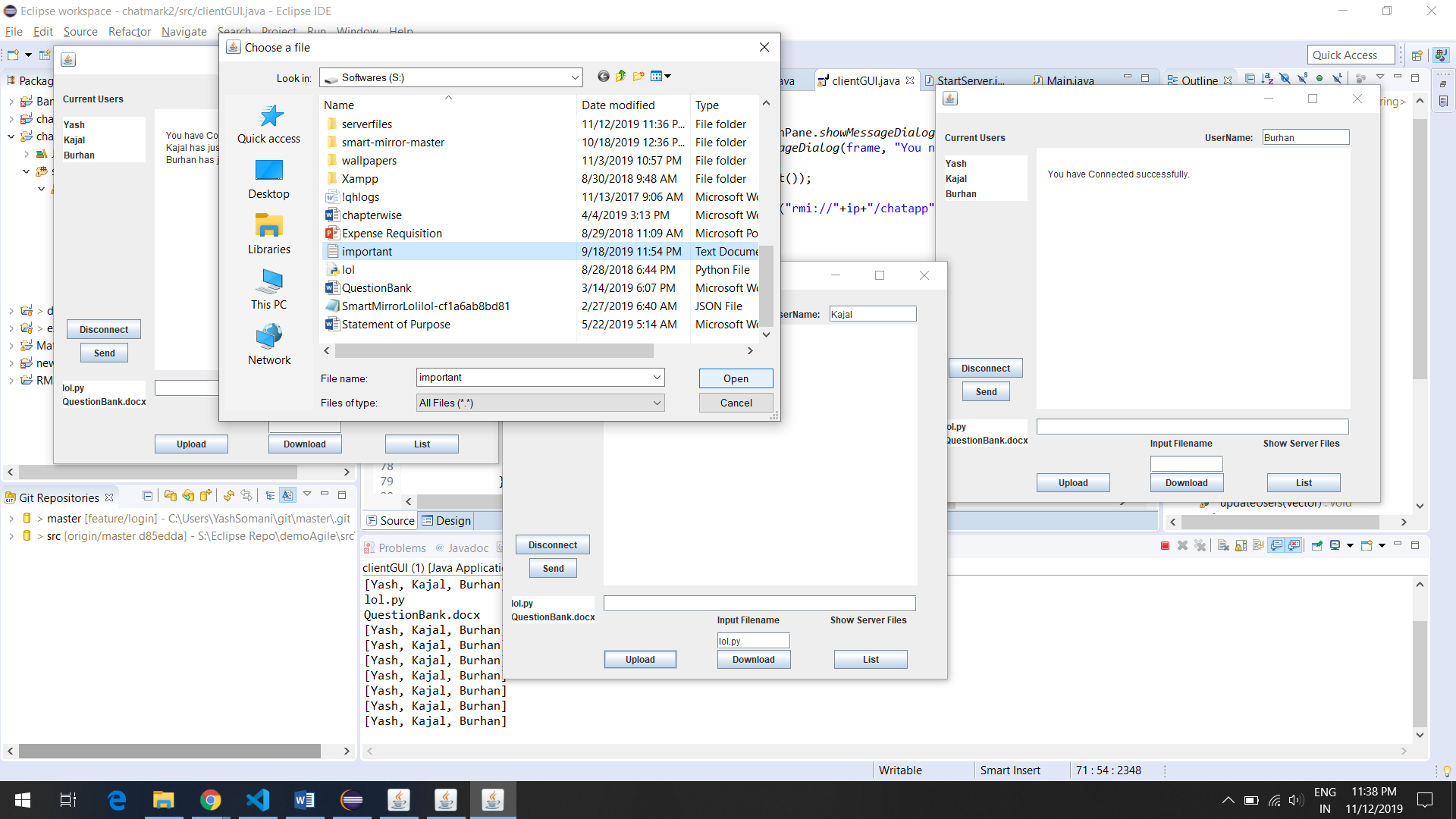
1. In the StartServer.java file, edit the IP address and mention your own IP Address where you want to start a server. Line(10).
2. In order to connect, you must enter your username first then click on start to connect to the server.
3. In the clientGUI.java file, edit the String ip and mention the IP address of the server. Line(45).
4. For upload functionality, you must do the following. Go to line 210. Edit the serverpath String in the quotes and mention the location of your own server files directory.
5. For download functionality, you must edit the locations of the clientpath and the serverpath in lines 258,259.
6. For listing down the files on the server, edit the server files directory on line 284.

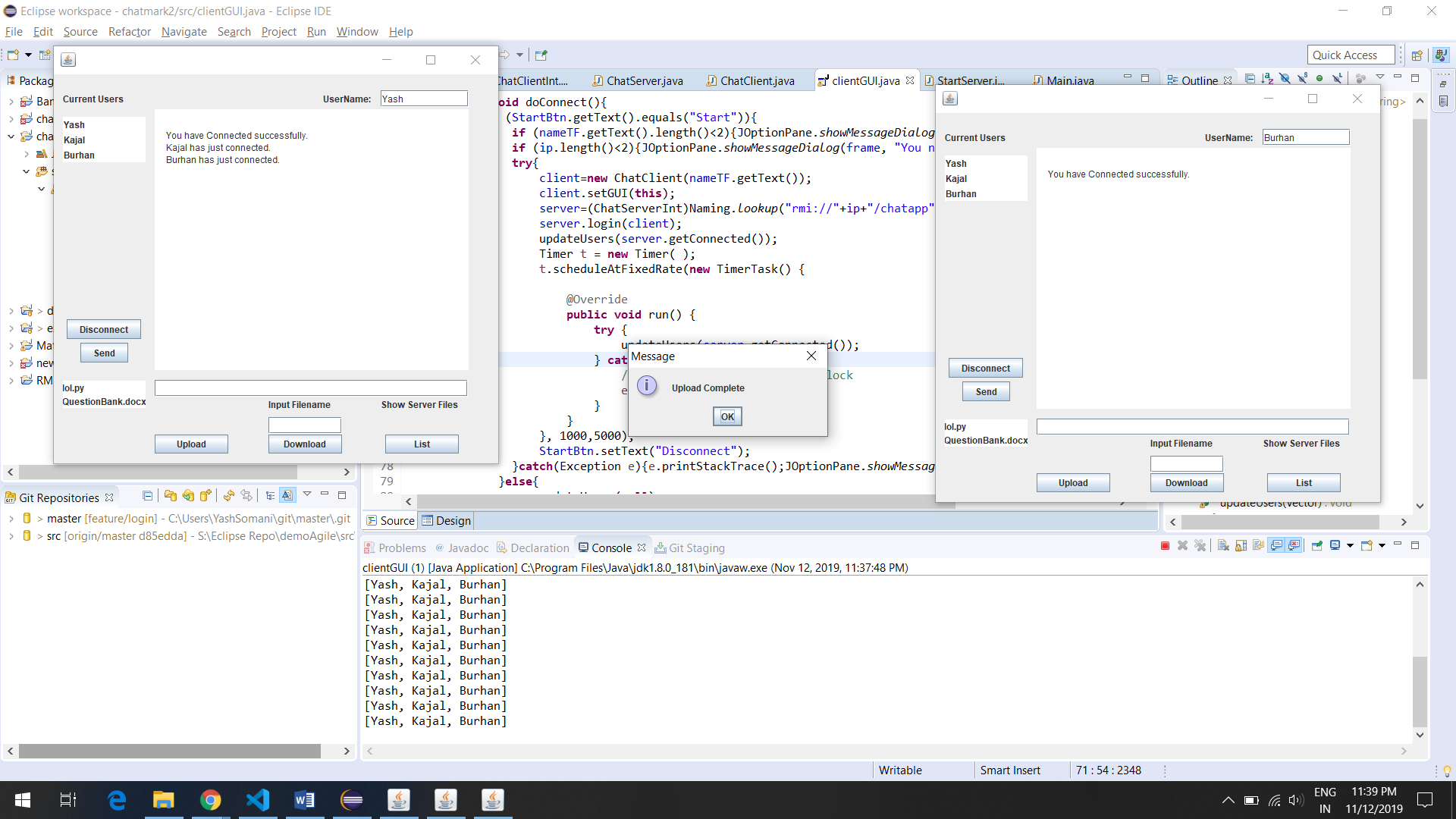
This project has used many GUI components, collections JList and Vectors, FileDialogBox, Dialog box, etc. The only difficultly faced was to update the clients GUI whenever anybody new connects to the system.

Some Output Snippets









Now if you check the list of the files on the server,

