

File Transfer and Chat Project in Java

Developed for the BSNL company, this File Transfer and Chat Project is a simple application developed using Java programming language. It aims at sharing information between exchanges of the company. Here, working on listening mode, client can [chat](#) or send message to other clients using signals. A single or multiple clients can be selected to send/receive files; it solely depends on whether or not the path exists.

Both file transferring and chatting (message sending) has been incorporated in a single system in this project. You can download the complete Java source code, project report, project files and documentation of File Transfer and Chat application from the download link in this post. Below, I've briefly introduced the features and various requirements of the system.

File Transfer and Chat Project Abstract:

This File Transfer and Chat (Message Sending) system has been developed in Java 1.3 which is based on Object Oriented Methodology. There are several packages in Java, but mainly swing packages and networking has been utilized in developing this project.

Modules Overview:

This project works under two modules, namely Active and Passive. Only passive clients can receive files, but active clients can send as well as receive files. Upon successful transfer of a file, an acknowledgement of successful file transfer is received, and then the passive clients can be disconnected. Any kind of files, including .fmx files, .exe files and more, can be sent using this system.

Features of Java:

- **Simple:** Java is a language which is based on Object Oriented Methodology, so it is very easy to learn and can be used effectively.
- **Robust:** – Java Programs are said to be robust because they will take care of memory management and will never crash under any circumstances.
- **Secure:** – Even though Java is developed using Object Oriented Principles, it eliminated the Pointers Concept. So it is not possible to access memory

directly, that's why Java is said to be Secure and is applicable for Internet, for that Applet is designed which can be understandable by the browsers.

- **Portable:** – Java Programs are Portable that those can be run under any kind of environment irrespective of the hardware used. This is known as platform independent.
- **Compiled & Interpreted:** – Unlike other Programming languages, Java code is both Compiled and Interpreted. The output after compilation is 'Byte Code' which is interpreted to produce output. This Byte Code is a new evolution, which makes Java a Platform Independent Language.

Since the File Transfer and Chat system is built using Objected Oriented Paradigm, the requirements of the clients are shown as Use cases and Scenarios. The detailed information for this can be found in the project report.

System Requirements:

1. Functional Requirements:

- It is only possible to connect to the server whenever it is under running. If the server is running, then any number of clients can get connected. Once a client connected to the server, it is not possible to connect again to the server.
- In order to send a file, the Source & Destination paths must be entered correctly. If the Source path is entered wrong or not entered, then an error should be displayed. If the Destination path is wrong then the client has to be intimated by the server that the path doesn't exist in the remote client.
- To disconnect a single [client](#) or multiple clients, choose the clients and disconnect. After disconnecting, observe whether the clients are in the connected clients list or not.

2. Performance Requirements:

- The Performance of the [Server](#) is not constant. It is depend upon the network constraints, the hardware configuration of the Server. If less than 50 clients are connected to the Server then the burden on server is less. If more than 50 clients are connected then the burden will increase.

- The remedy to the problem is simply disconnecting some of the clients in passive state. The response time will also be considered to assess performance, which is also depending upon the network constraints, and the hardware configuration of Server.

3. Server-side Hardware Requirements:

- Console: – Mouse, Monitor, 104keys Keyboard
- Processor: – Intel Pentium III processor, 1.1 GHz speed.
- Memory: – 128MB RAM, 20GB Hard Disk.
- N/w Components: – Network Adapter, RJ-45 Connector, HUB.

4. Client-side Hardware Requirements:

- Console: – Mouse, Monitor, 104keys Keyboard
- Processor: – Intel Pentium III processor.
- Memory: – 64 MB RAM, 20GB Hard Disk.
- N/w Components: – Network Adapter, RJ-45 Connector, UTP Cable, HUB.

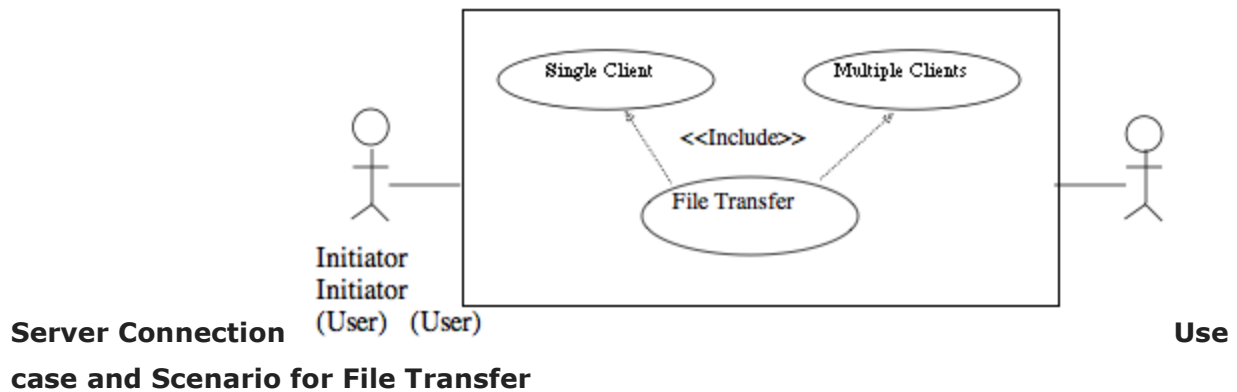
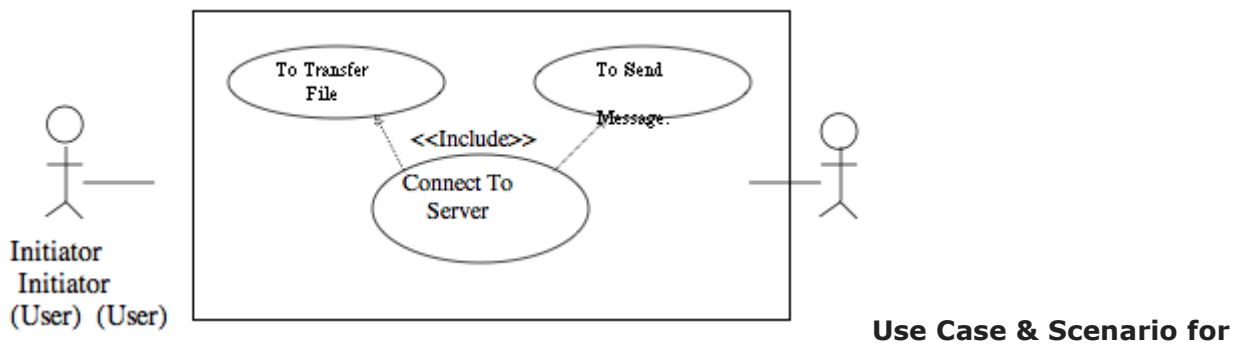
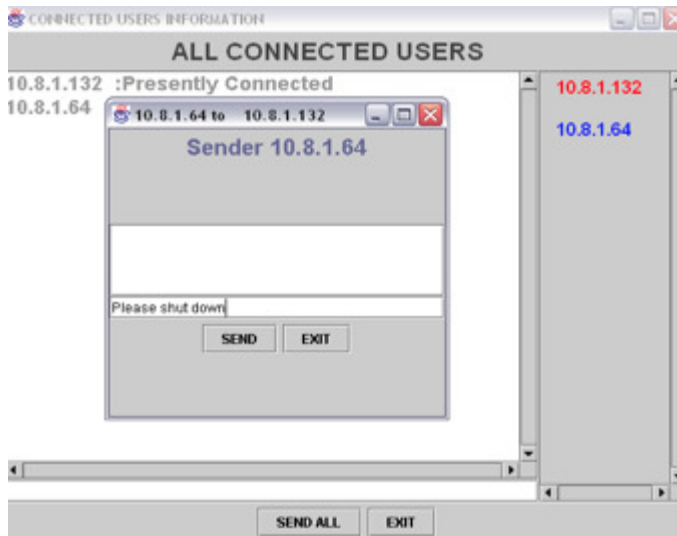
5. Server-side Software Requirements:

- Operating System: – MS-DOS, Windows 9x/XP (or) Windows NT.
- System Software : – JDK 1.2.2 (or) JDK 1.3 (or) JDK 1.4.
- Application Software: – Ftpser.class

6. Client-side Software Requirements:

- Operating System: – MS-DOS, Windows 9x.
- Database: – Oracle.
- System Software : – JDK 1.2.2 (or) JDK 1.3 (or) JDK 1.4.
- Application Software: – ActiveForm.class, ChatForm.class, Sendall.class, Killall.class, Form.class, Filter.class, Chat.class

Screenshots/Diagrams:



Conclusion:

This File Transfer and Chat Project in Java can be successfully operated in real environment and can be tested practically. The files and chat messages can be sent to the selected clients. Even .exe files can be transferred correctly without any errors and are perfectly executed upon download.

