Manual :

System Directories and contents:
- Source: Contains all source file for server and client
                Contains Ant build file
- execution: execution files including ClientFinal.jar and ServerFinal.jar
                Shared directory includes FileGenerator.java for creating txt files.
- OutputFile: Outputfiles of server and peer execution log
- nun.sh: run bash script to execute system

Source Code Import
Import Prog_Ass_1 directory from Source folder into Eclipse IDE .
Then build project using Ant build

Setup System Environment:

- Go to execution directory.
- Run FileGenerator.java from peer1, peer2, peer3 directory to create 1000 txt files of different size.

$ javac FileGenerator.java
$ java FileGenerator

Execution of System:

For Server
- Run run.sh bash script for exection

$ bash run.sh

- It will prompt you to select 1 for server and 2 for client
  'Enter 1 to start server'

  Please enter Port No : <Enter port number for server to start>
- It display below message after successfully starting of server

  Server running on IP : 192.168.0.23.......... Port No : 8888Trying to connect

For Peer
- Run run.sh bash script for execution

$ bash run.sh

- It will prompt you to select 1 for server and 2 for client
  'Enter 2 to start server'

<Display Client IP Address>

Please enter Port No for Peer to Peer connection :
<Enter Port for Peer Server for P2P>

Please enter Shared Directory Name :
<Enter directory to share>

Please enter Ip Addess For socket connection :
<Enter Indexing Server IP address>
Waiting...

Please enter Port No for Scoket connection :
<Enter Indexing Server port to connec>
<Display NO of Files Registered>
Register Response time: 41210000ns

PeerID: 8880
Options:
1 - Search for filename
2 - Obtain filename from peer
3 - List files in shared directory
4 - Exit

<select option> : "1"
Enter filename: <Enter Filename to lookup>

Below Peer have 2000.txt in their shared directory :
192.168.0.23:8870
Lookup Response time: 4616000 ns

PeerID: 8880
Options:
1 - Search for filename
2 - Obtain filename from peer
3 - List files in shared directory
4 - Exit

<select option> : "2"
Enter filename:
<Enter Filename to obtain>
Enter Peer Server IP Address:
<Enter Peer IP from where file will be downloaded>
Enter Peer Server Port No:
<Enter port no of peer server>

Run From jar files

--Goto execution directory and run ServerFinal.jar

> *$ java –jar ServerFinal.jar*

-Run ClienFinal.jar as many times you want # of Peers

> *$ java –jar ClientFInal.jar*