GNUTELLA STYLE PEER TO PEER FILE SHARING

by
Abdullah Ejaz

MAY 3, 2018

TEXAS TECH UNIVERSITY
Lubbock, Texas

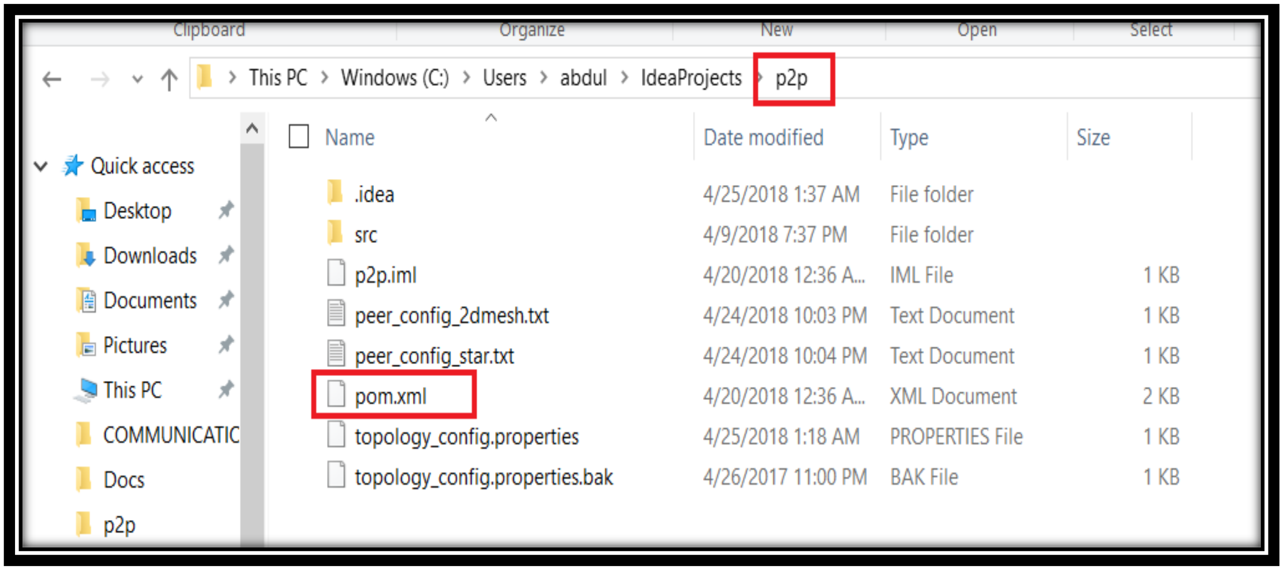
Table of Contents

A. Extracting the folder

- Unzip the project folder and extract all the files.
- Then perform the following step to run the application step by step.

B. Building the Application

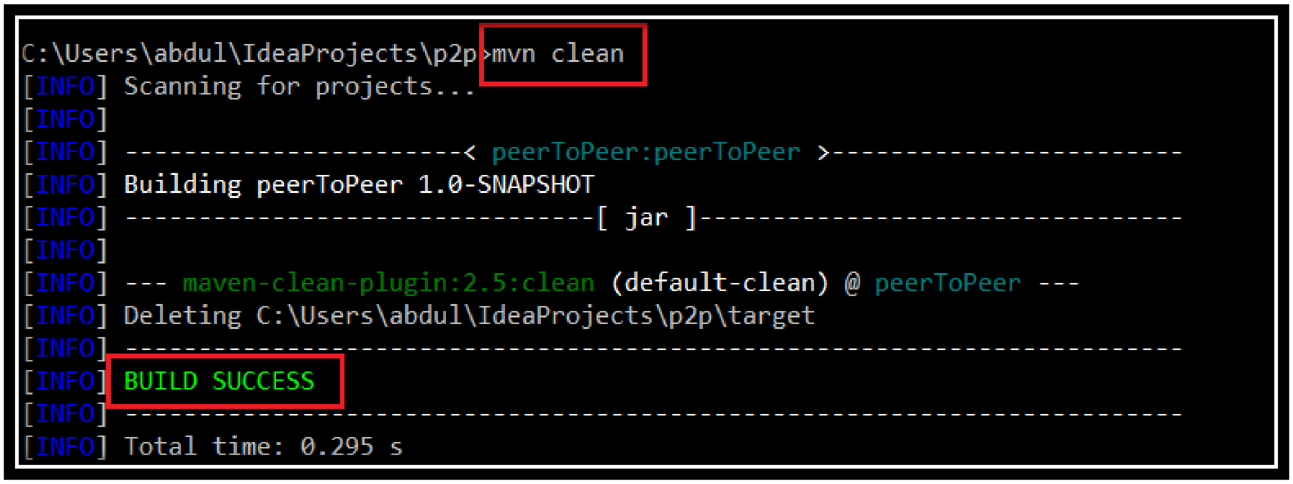
- Open Command Prompt.
- Go to the root folder of the project i.e. “**p2p**” where **pom.xml** file is located.



Name	Date modified	Type	Size
.idea	4/25/2018 1:37 AM	File folder	
src	4/9/2018 7:37 PM	File folder	
p2p.iml	4/20/2018 12:36 A...	IML File	1 KB
peer_config_2dmesh.txt	4/24/2018 10:03 PM	Text Document	1 KB
peer_config_star.txt	4/24/2018 10:04 PM	Text Document	1 KB
pom.xml	4/20/2018 12:36 A...	XML Document	2 KB
topology_config.properties	4/25/2018 1:18 AM	PROPERTIES File	1 KB
topology_config.properties.bak	4/26/2017 11:00 PM	BAK File	1 KB

Fig 1. Project root folder (pom.xml location)

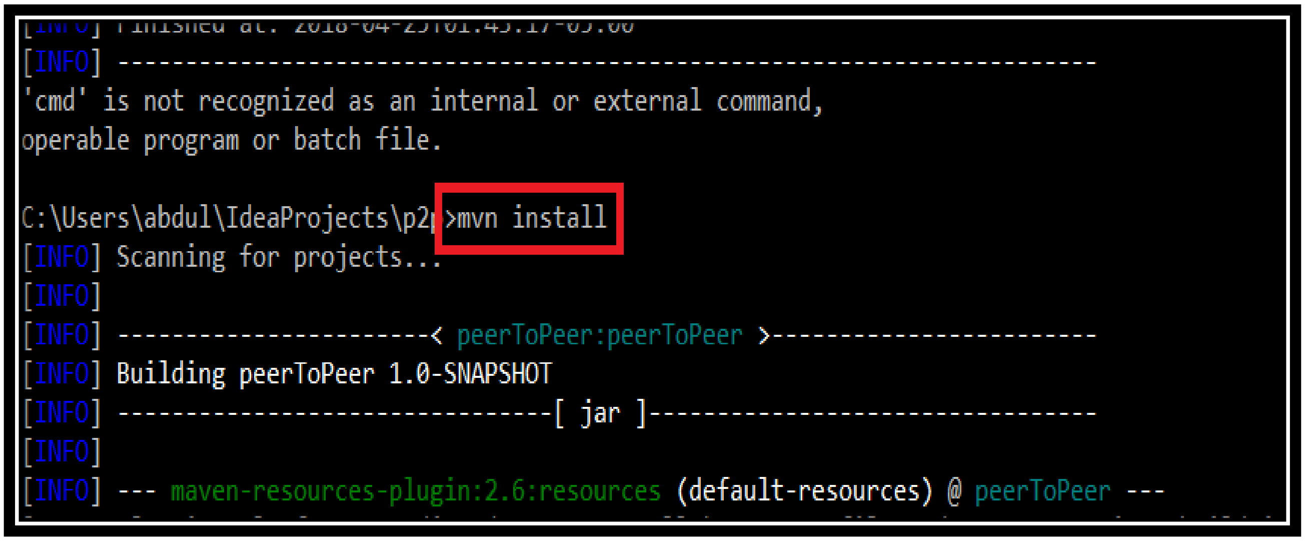
- Type the command **mvn clean** in **cmd** and press enter.



```
C:\Users\abdul\IdeaProjects\p2p>mvn clean
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< peerToPeer:peerToPeer >------------------------
[INFO] Building peerToPeer 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ peerToPeer ---
[INFO] Deleting C:\Users\abdul\IdeaProjects\p2p\target
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 0.295 s
```

Fig 2. Console output for Command mvn clean

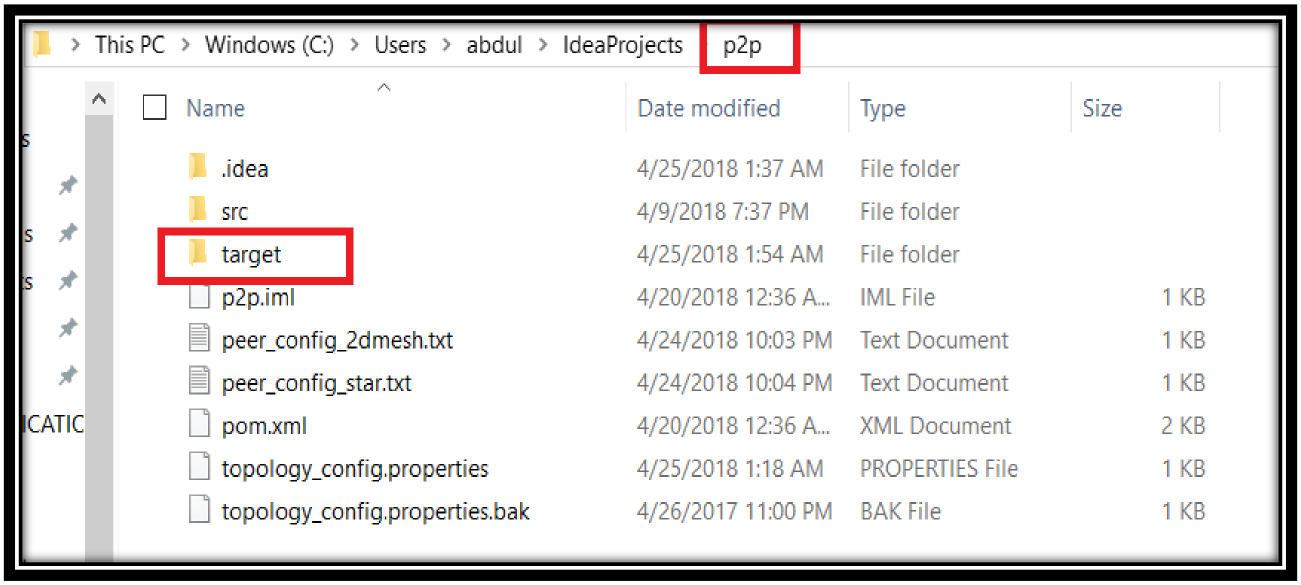
- Type the command **mvn install** in **cmd** and press Enter.



```
[INFO] Finished at: 2018-04-25T01:45:17-05:00
[INFO] ------------------------------------------------------------------------
'cmd' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\abdul\IdeaProjects\p2p>mvn install
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< peerToPeer:peerToPeer >------------------------
[INFO] Building peerToPeer 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ peerToPeer ---
```

Fig 3. Console output for Command mvn install

- This will create a “**target**” folder in the directory and **p2p.jar** file in the newly created target folder.

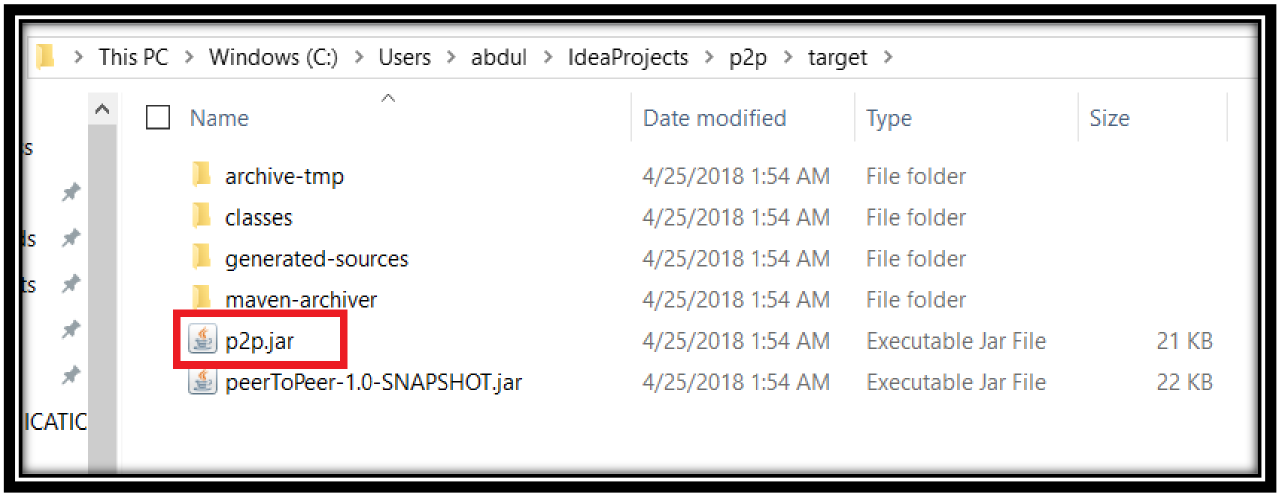


This PC > Windows (C:) > Users > abdul > IdeaProjects > p2p

Name	Date modified	Type	Size
.idea	4/25/2018 1:37 AM	File folder	
src	4/9/2018 7:37 PM	File folder	
target	4/25/2018 1:54 AM	File folder	
p2p.iml	4/20/2018 12:36 A...	IML File	1 KB
peer_config_2dmesh.txt	4/24/2018 10:03 PM	Text Document	1 KB
peer_config_star.txt	4/24/2018 10:04 PM	Text Document	1 KB
pom.xml	4/20/2018 12:36 A...	XML Document	2 KB
topology_config.properties	4/25/2018 1:18 AM	PROPERTIES File	1 KB
topology_config.properties.bak	4/26/2017 11:00 PM	BAK File	1 KB

Fig 4. Target folder created after executing mvn install

- Go to the target folder by typing **cd target** in **cmd** (.jar file will be in target folder).



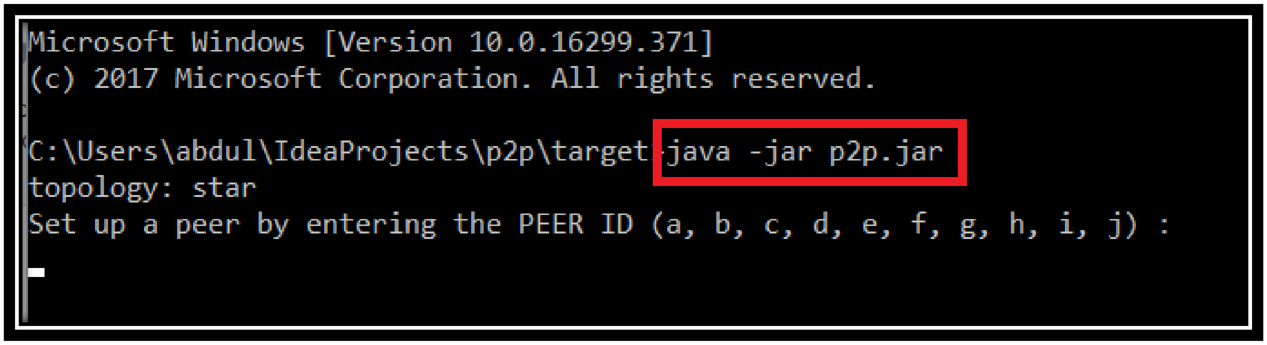
This PC > Windows (C:) > Users > abdul > IdeaProjects > p2p > target >

Name	Date modified	Type	Size
archive-tmp	4/25/2018 1:54 AM	File folder	
classes	4/25/2018 1:54 AM	File folder	
generated-sources	4/25/2018 1:54 AM	File folder	
maven-archiver	4/25/2018 1:54 AM	File folder	
p2p.jar	4/25/2018 1:54 AM	Executable Jar File	21 KB
peerToPeer-1.0-SNAPSHOT.jar	4/25/2018 1:54 AM	Executable Jar File	22 KB

Fig 5. Jar File created after mvn install

C. Running the Program

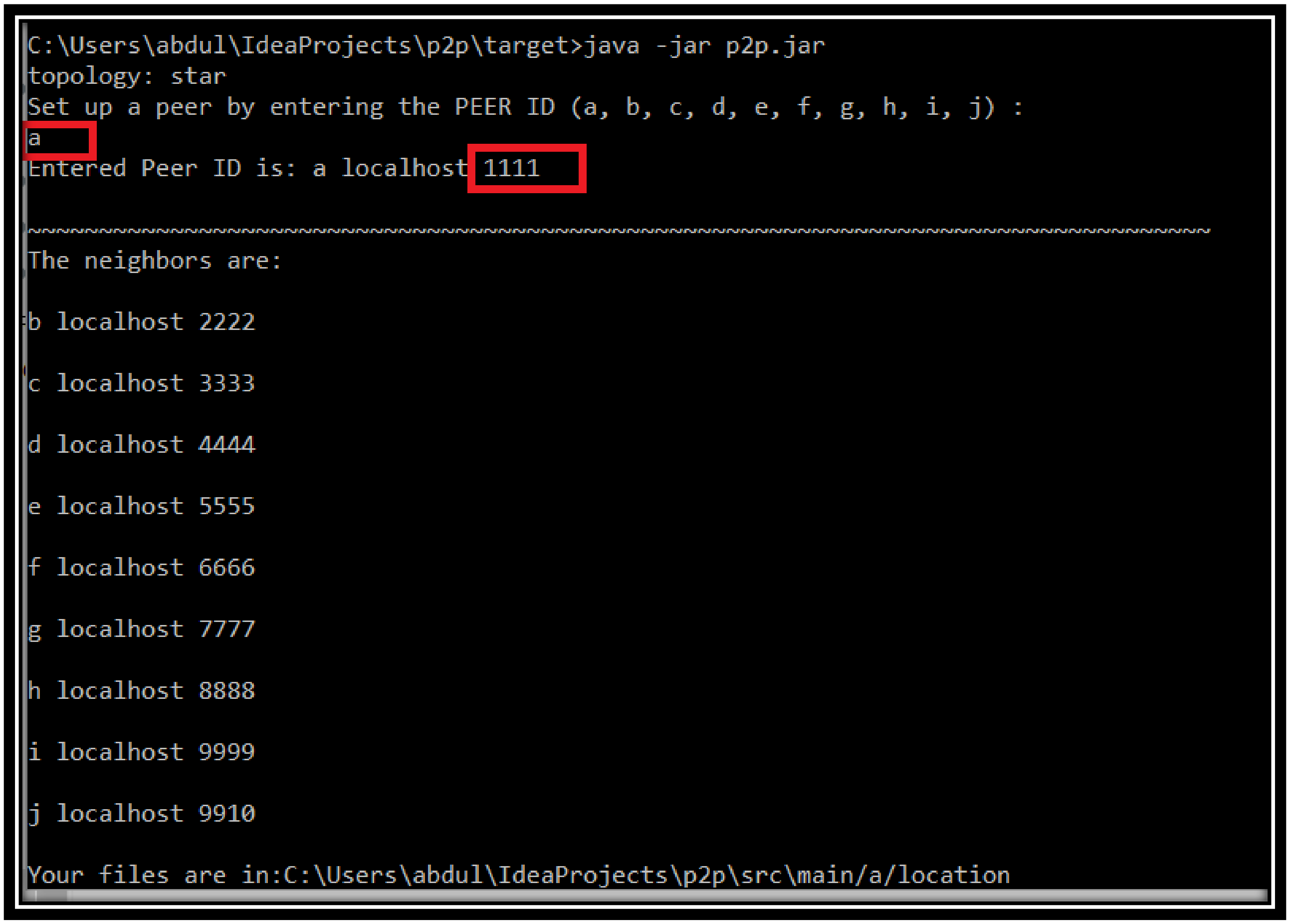
- Type the command java -jar p2p.jar in the **cmd** and press enter.



```
Microsoft Windows [Version 10.0.16299.371]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\abdul\IdeaProjects\p2p\target>java -jar p2p.jar
topology: star
Set up a peer by entering the PEER ID (a, b, c, d, e, f, g, h, i, j) :
```

Fig 6. Running the Jar File in the Console

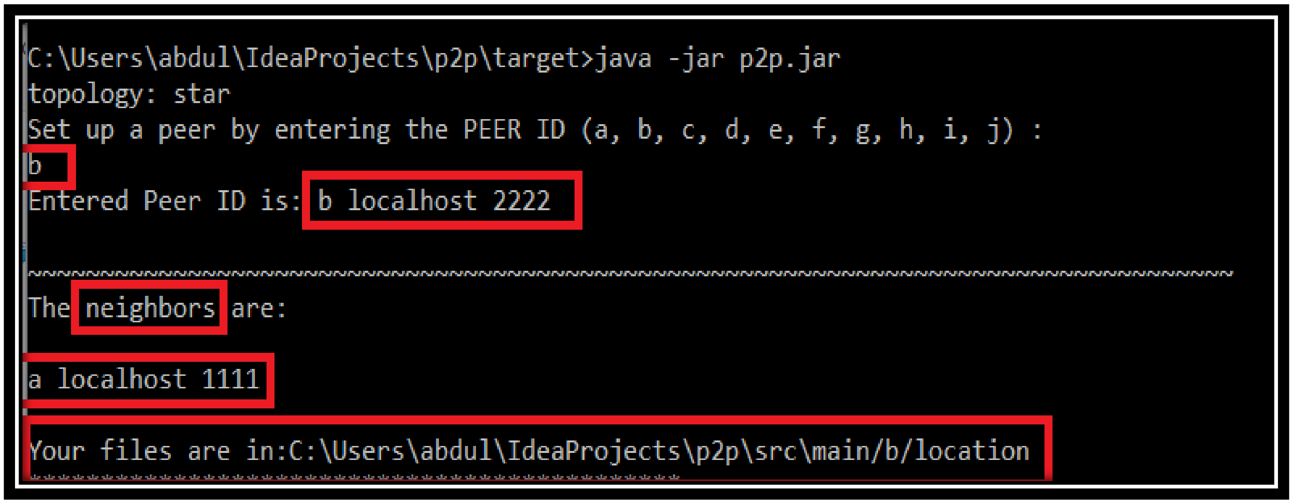
- Now to start a peer, enter a **peer ID** from the choices given. For ex. **"a"**



```
C:\Users\abdul\IdeaProjects\p2p\target>java -jar p2p.jar
topology: star
Set up a peer by entering the PEER ID (a, b, c, d, e, f, g, h, i, j) :
a
Entered Peer ID is: a localhost 1111

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
The neighbors are:

b localhost 2222

c localhost 3333

d localhost 4444

e localhost 5555

f localhost 6666

g localhost 7777

h localhost 8888

i localhost 9999

j localhost 9910

Your files are in:C:\Users\abdul\IdeaProjects\p2p\src\main/a/location
```

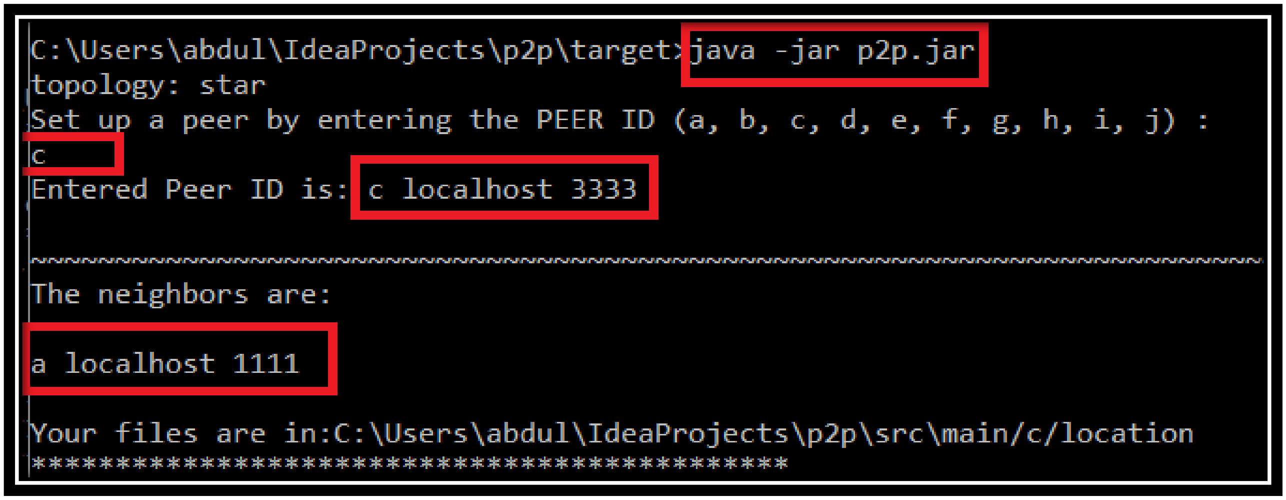
Fig 7. Peer 1 started after giving command

- Similarly **repeat** the process for all the 10 peers.



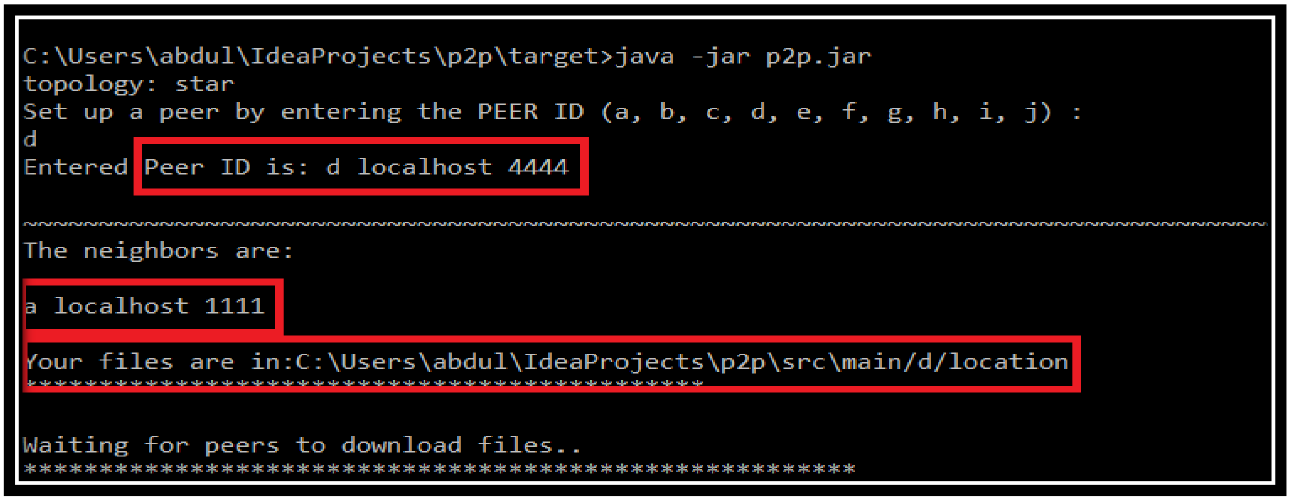
```
C:\Users\abdul\IdeaProjects\p2p\target>java -jar p2p.jar
topology: star
Set up a peer by entering the PEER ID (a, b, c, d, e, f, g, h, i, j) :
b
Entered Peer ID is: b localhost 2222

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
The neighbors are:

a localhost 1111

Your files are in:C:\Users\abdul\IdeaProjects\p2p\src\main/b/location
*********************************************
```

Fig 8. Peer 2 started after selecting option b



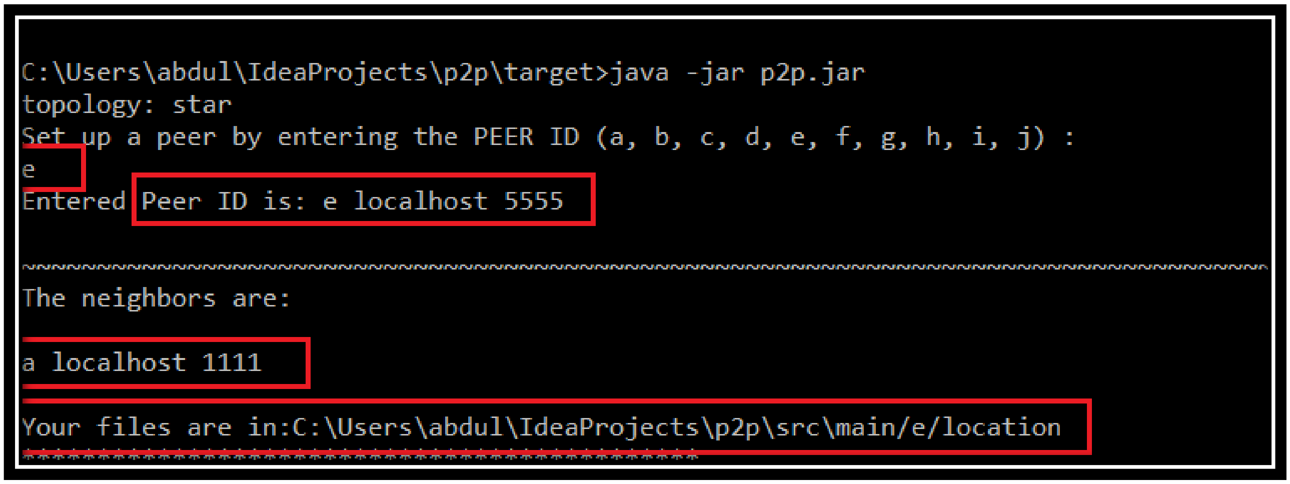
```
C:\Users\abdul\IdeaProjects\p2p\target>java -jar p2p.jar
topology: star
Set up a peer by entering the PEER ID (a, b, c, d, e, f, g, h, i, j) :
c
Entered Peer ID is: c localhost 3333

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
The neighbors are:

a localhost 1111

Your files are in:C:\Users\abdul\IdeaProjects\p2p\src\main/c/location
*********************************************
```

Fig 9. Peer 3 started after selecting option c



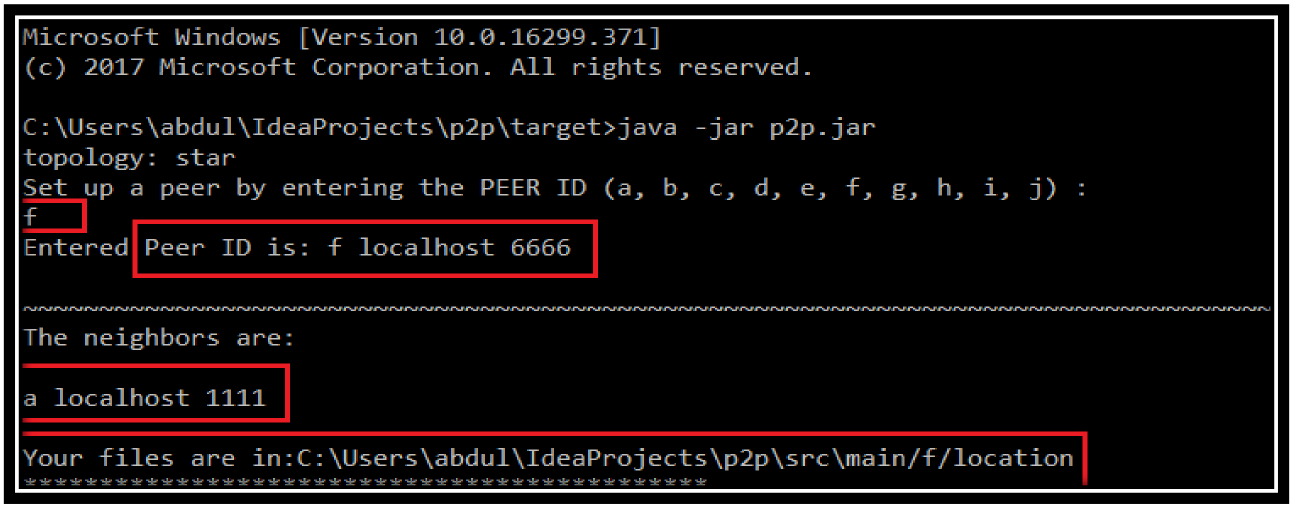
```
C:\Users\abdul\IdeaProjects\p2p\target>java -jar p2p.jar
topology: star
Set up a peer by entering the PEER ID (a, b, c, d, e, f, g, h, i, j) :
d
Entered Peer ID is: d localhost 4444

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
The neighbors are:

a localhost 1111

Your files are in:C:\Users\abdul\IdeaProjects\p2p\src\main/d/location
*********************************************

Waiting for peers to download files..
*******************************************************
```

Fig 10. Peer 4 started after selecting option d



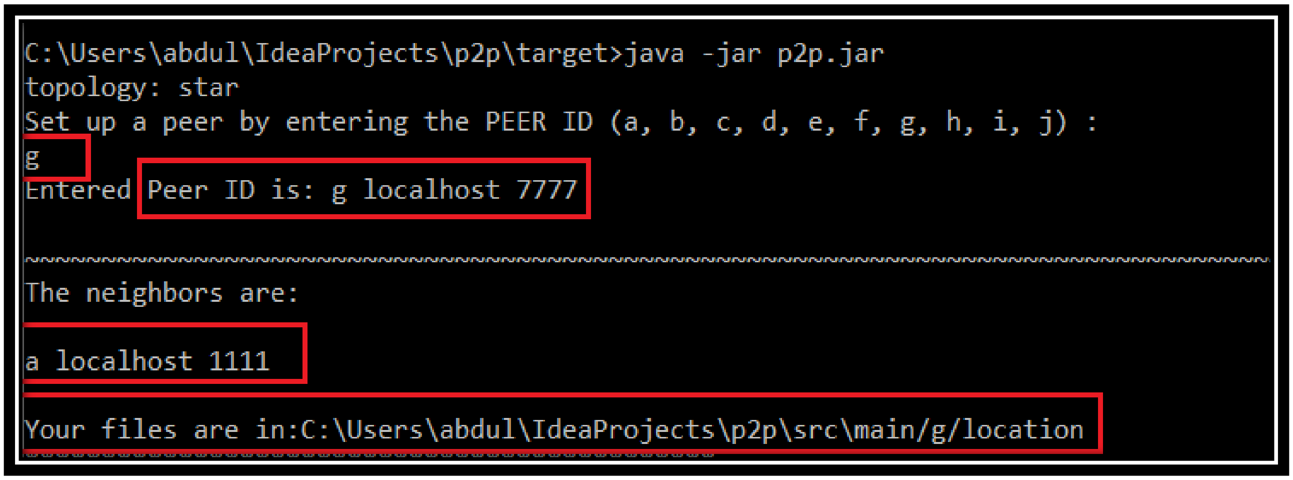
```
C:\Users\abdul\IdeaProjects\p2p\target>java -jar p2p.jar
topology: star
Set up a peer by entering the PEER ID (a, b, c, d, e, f, g, h, i, j) :
e
Entered Peer ID is: e localhost 5555

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
The neighbors are:

a localhost 1111

Your files are in:C:\Users\abdul\IdeaProjects\p2p\src\main/e/location
*********************************************
```

Fig 11. Peer 5 started after selecting option e



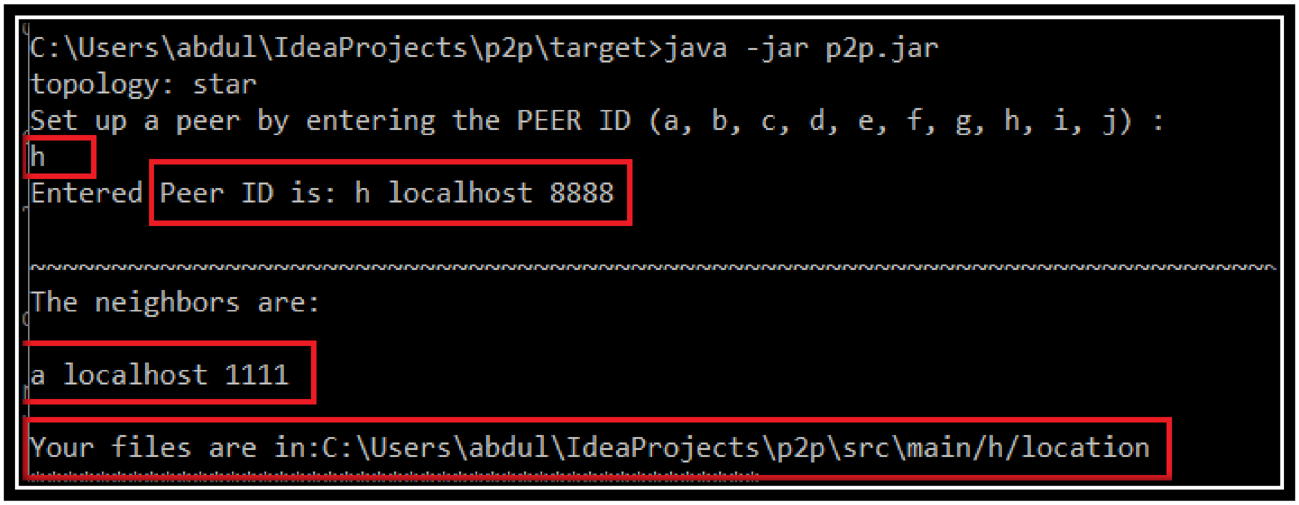
```
Microsoft Windows [Version 10.0.16299.371]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\abdul\IdeaProjects\p2p\target>java -jar p2p.jar
topology: star
Set up a peer by entering the PEER ID (a, b, c, d, e, f, g, h, i, j) :
f
Entered Peer ID is: f localhost 6666

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
The neighbors are:

a localhost 1111

Your files are in:C:\Users\abdul\IdeaProjects\p2p\src\main/f/location
*********************************************
```

Fig 12. Peer 6 started after selecting option f



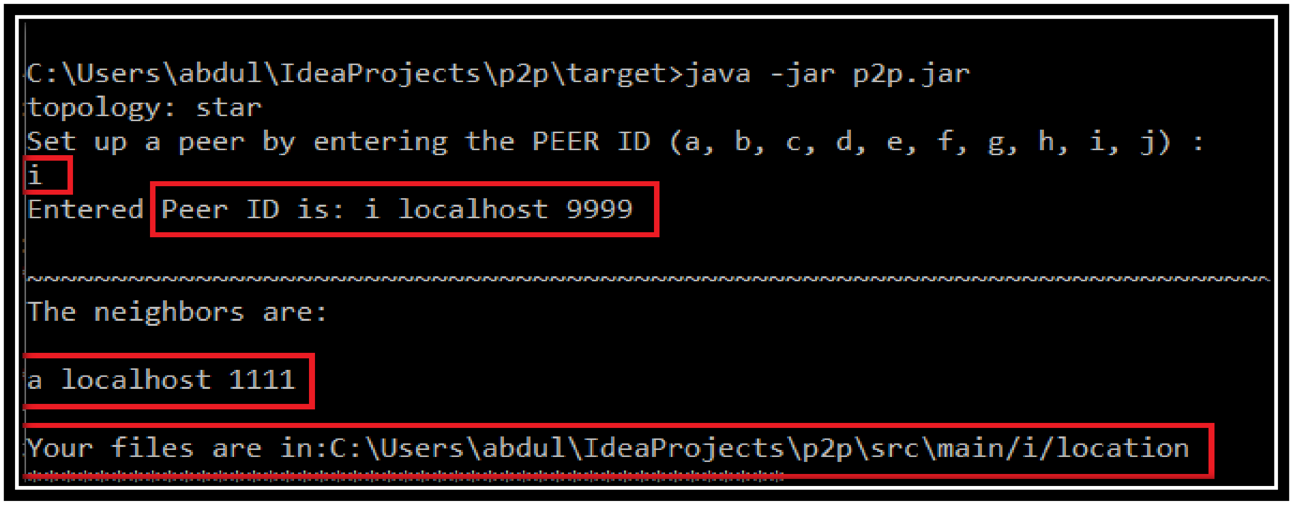
```
C:\Users\abdul\IdeaProjects\p2p\target>java -jar p2p.jar
topology: star
Set up a peer by entering the PEER ID (a, b, c, d, e, f, g, h, i, j) :
g
Entered Peer ID is: g localhost 7777

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
The neighbors are:

a localhost 1111

Your files are in:C:\Users\abdul\IdeaProjects\p2p\src\main/g/location
```

Fig 13. Peer 7 started after selecting option g



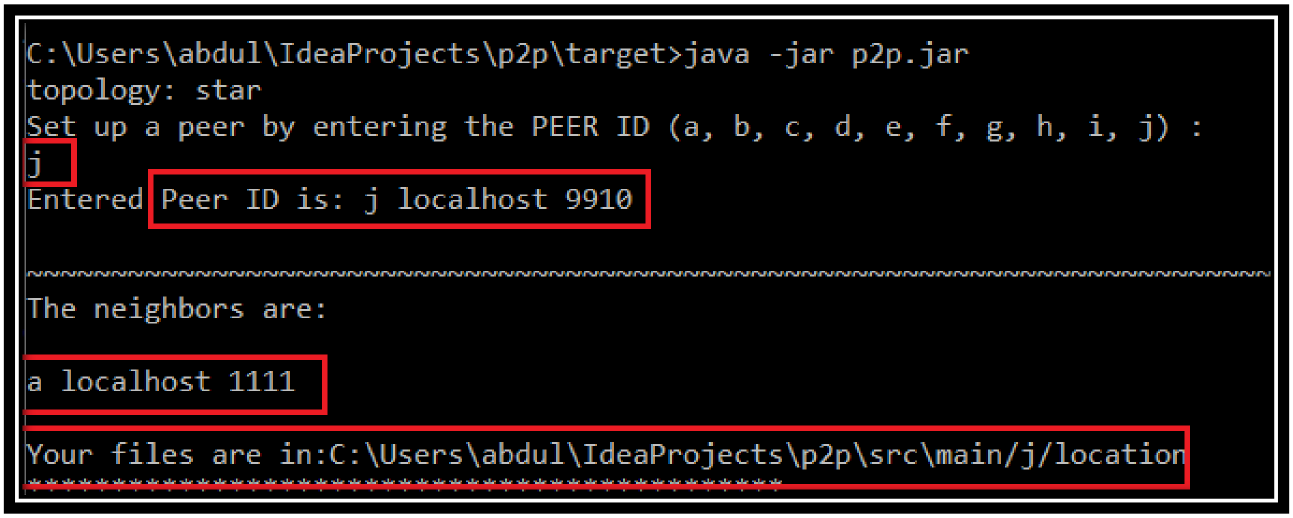
```
C:\Users\abdul\IdeaProjects\p2p\target>java -jar p2p.jar
topology: star
Set up a peer by entering the PEER ID (a, b, c, d, e, f, g, h, i, j) :
h
Entered Peer ID is: h localhost 8888

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
The neighbors are:

a localhost 1111

Your files are in:C:\Users\abdul\IdeaProjects\p2p\src\main/h/location
```

Fig 14. Peer 8 started after selecting option h



```
C:\Users\abdul\IdeaProjects\p2p\target>java -jar p2p.jar
topology: star
Set up a peer by entering the PEER ID (a, b, c, d, e, f, g, h, i, j) :
i
Entered Peer ID is: i localhost 9999

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
The neighbors are:

a localhost 1111

Your files are in:C:\Users\abdul\IdeaProjects\p2p\src\main/i/location
```

Fig 15. Peer 9 started after selecting option i

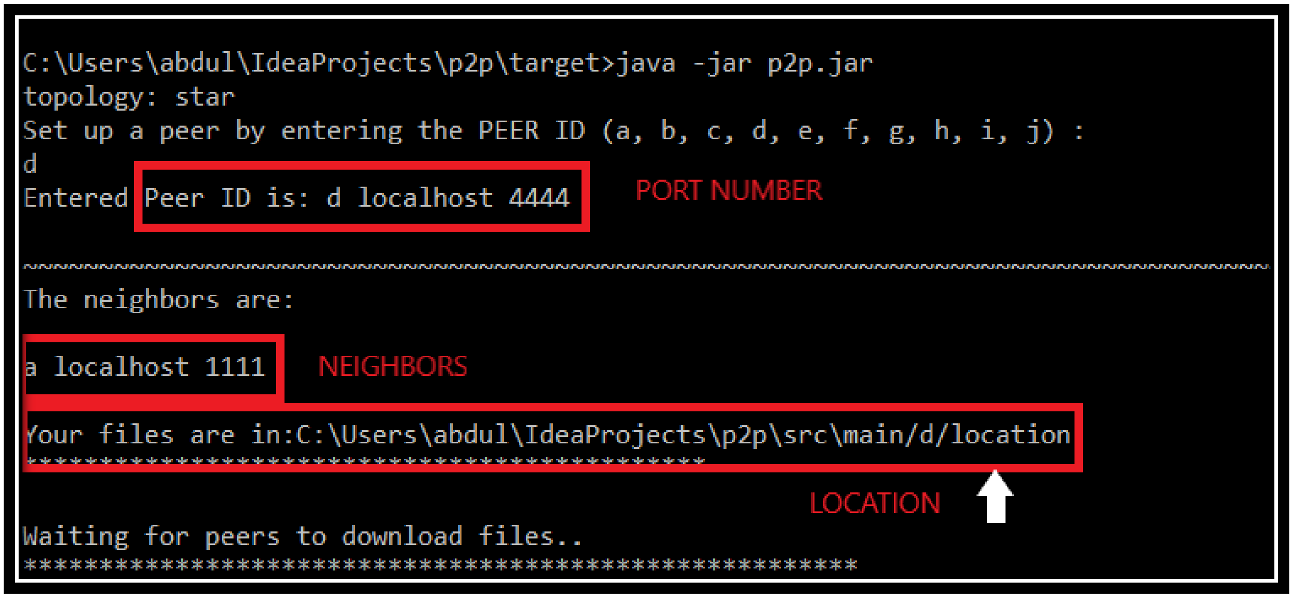


```
C:\Users\abdul\IdeaProjects\p2p\target>java -jar p2p.jar
topology: star
Set up a peer by entering the PEER ID (a, b, c, d, e, f, g, h, i, j) :
j
Entered Peer ID is: j localhost 9910

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
The neighbors are:

a localhost 1111

Your files are in:C:\Users\abdul\IdeaProjects\p2p\src\main/j/location
*************************************************
```

Fig 16. Peer 10 started after selecting option j

After starting all the peer, we can notice in all the screenshot there are some other highlighted information. Those information are:

- Port number of the Peer
- The neighbor peers &
- Location of the Peer



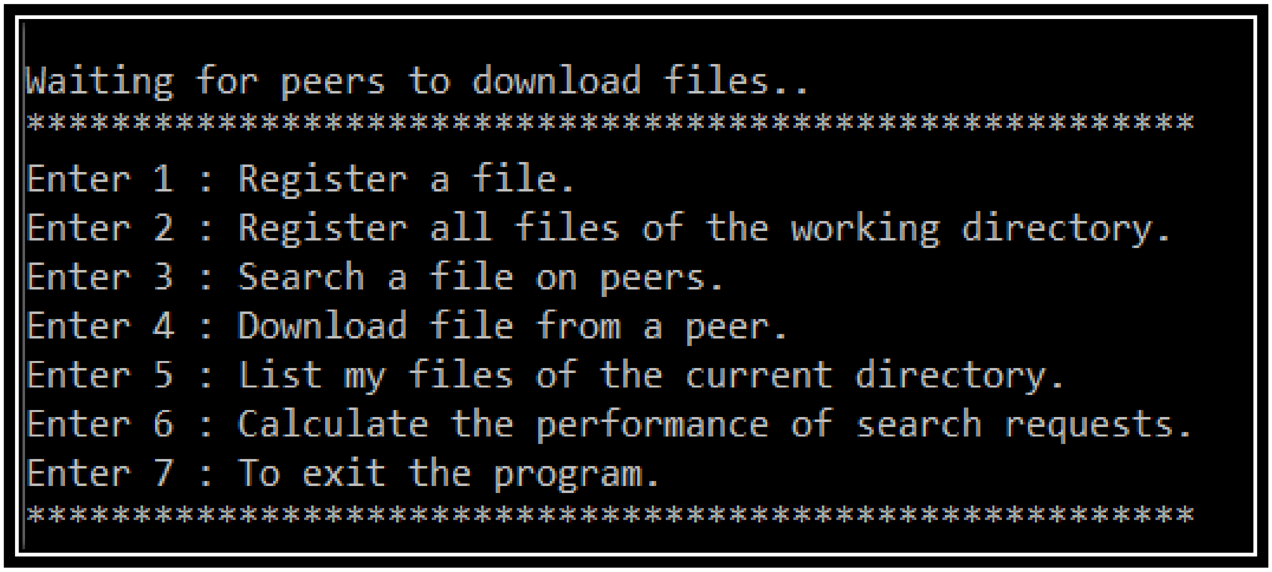
```
C:\Users\abdul\IdeaProjects\p2p\target>java -jar p2p.jar
topology: star
Set up a peer by entering the PEER ID (a, b, c, d, e, f, g, h, i, j) :
d
Entered Peer ID is: d localhost 4444

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
The neighbors are:

a localhost 1111

Your files are in:C:\Users\abdul\IdeaProjects\p2p\src\main/d/location
*************************************************

Waiting for peers to download files..
*******************************************************
```

Fig 17. Information on the console after running a Peer

D. Actions on Peers

After all the peers are up and running. There will appear a **main menu** in each peer console which will have all these functionalities.

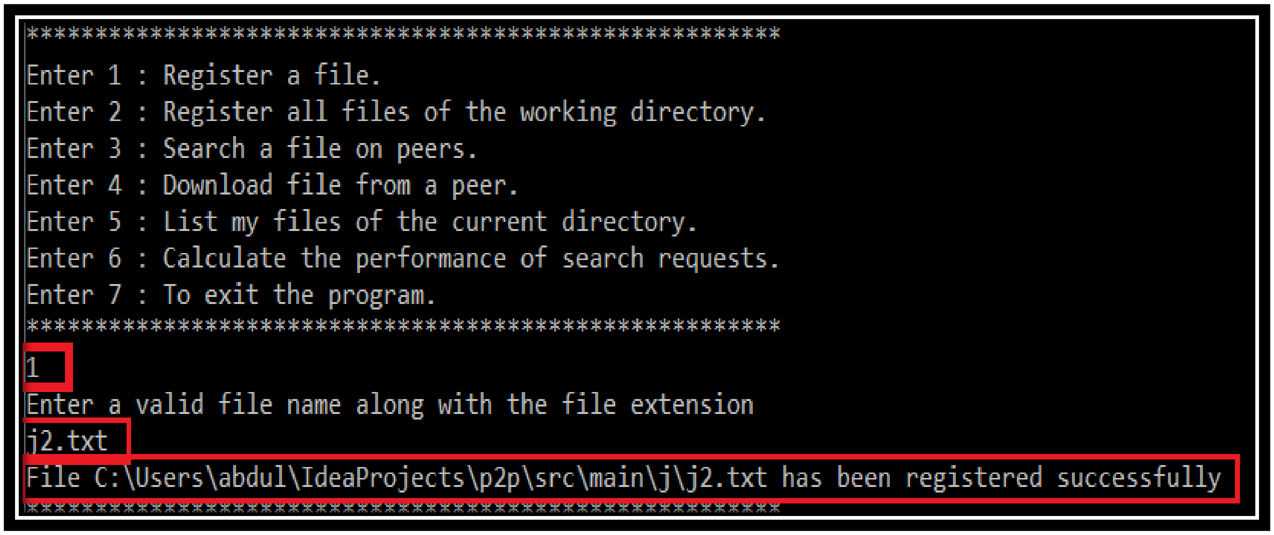
- Register a file
- Register all files of the working directory.
- Search a file on peers.
- Download file from peer.
- List my files of the current directory.
- Calculate the performance of search requests.
- Exit option



```
Waiting for peers to download files..
*******************************************************
Enter 1 : Register a file.
Enter 2 : Register all files of the working directory.
Enter 3 : Search a file on peers.
Enter 4 : Download file from a peer.
Enter 5 : List my files of the current directory.
Enter 6 : Calculate the performance of search requests.
Enter 7 : To exit the program.
*******************************************************
```

Fig 18. Actions that can be performed on the Peers.

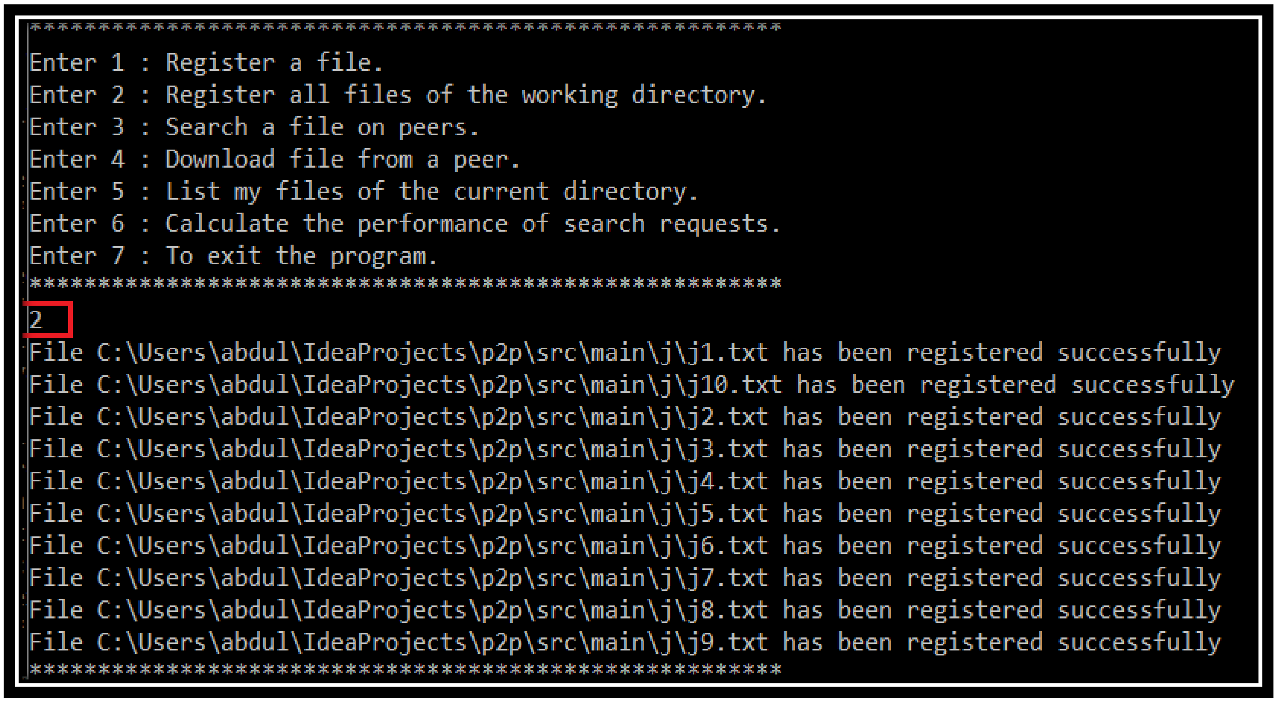
1. Register a file of a Peer



```
*******************************************************
Enter 1 : Register a file.
Enter 2 : Register all files of the working directory.
Enter 3 : Search a file on peers.
Enter 4 : Download file from a peer.
Enter 5 : List my files of the current directory.
Enter 6 : Calculate the performance of search requests.
Enter 7 : To exit the program.
*******************************************************
1
Enter a valid file name along with the file extension
j2.txt
File C:\Users\abdul\IdeaProjects\p2p\src\main\j\j2.txt has been registered successfully
*******************************************************
```

Figure 19. Registering a file on Peer

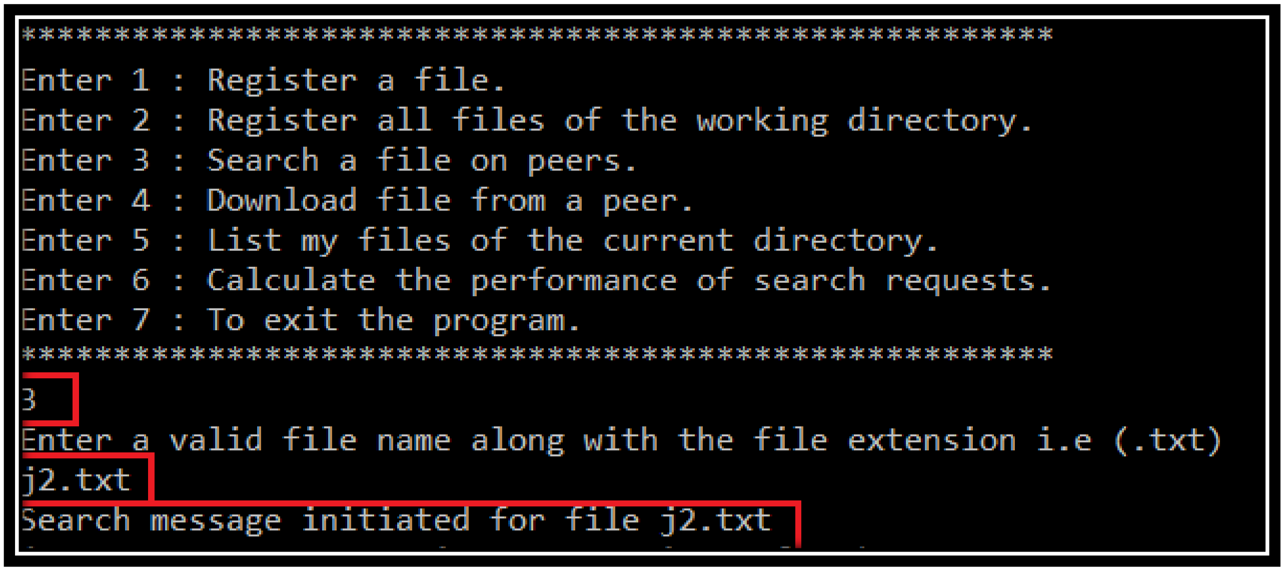
2. To register all files of a peer.



```
*******************************************************
Enter 1 : Register a file.
Enter 2 : Register all files of the working directory.
Enter 3 : Search a file on peers.
Enter 4 : Download file from a peer.
Enter 5 : List my files of the current directory.
Enter 6 : Calculate the performance of search requests.
Enter 7 : To exit the program.
*******************************************************
2
File C:\Users\abdul\IdeaProjects\p2p\src\main\j\j1.txt has been registered successfully
File C:\Users\abdul\IdeaProjects\p2p\src\main\j\j10.txt has been registered successfully
File C:\Users\abdul\IdeaProjects\p2p\src\main\j\j2.txt has been registered successfully
File C:\Users\abdul\IdeaProjects\p2p\src\main\j\j3.txt has been registered successfully
File C:\Users\abdul\IdeaProjects\p2p\src\main\j\j4.txt has been registered successfully
File C:\Users\abdul\IdeaProjects\p2p\src\main\j\j5.txt has been registered successfully
File C:\Users\abdul\IdeaProjects\p2p\src\main\j\j6.txt has been registered successfully
File C:\Users\abdul\IdeaProjects\p2p\src\main\j\j7.txt has been registered successfully
File C:\Users\abdul\IdeaProjects\p2p\src\main\j\j8.txt has been registered successfully
File C:\Users\abdul\IdeaProjects\p2p\src\main\j\j9.txt has been registered successfully
*******************************************************
```

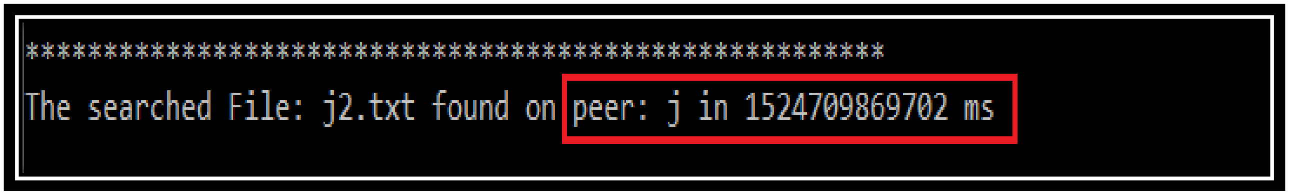
Figure 20. Registering all files of the Peer

3. Search a file on Peers



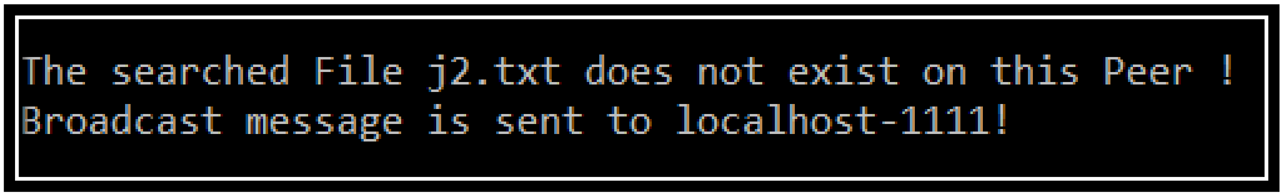
```
*******************************************************
Enter 1 : Register a file.
Enter 2 : Register all files of the working directory.
Enter 3 : Search a file on peers.
Enter 4 : Download file from a peer.
Enter 5 : List my files of the current directory.
Enter 6 : Calculate the performance of search requests.
Enter 7 : To exit the program.
*******************************************************
3
Enter a valid file name along with the file extension i.e (.txt)
j2.txt
Search message initiated for file j2.txt
```

- After the Peer found the File, the following message will appear.



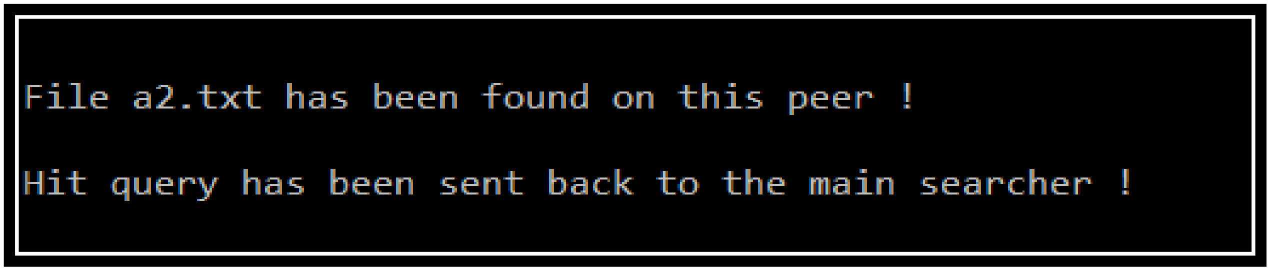
```
**********************************************************
The searched File: j2.txt found on peer: j in 1524709869702 ms
```

- All the other peers except Peer J will show this message on the console.



```
The searched File j2.txt does not exist on this Peer !
Broadcast message is sent to localhost-1111!
```

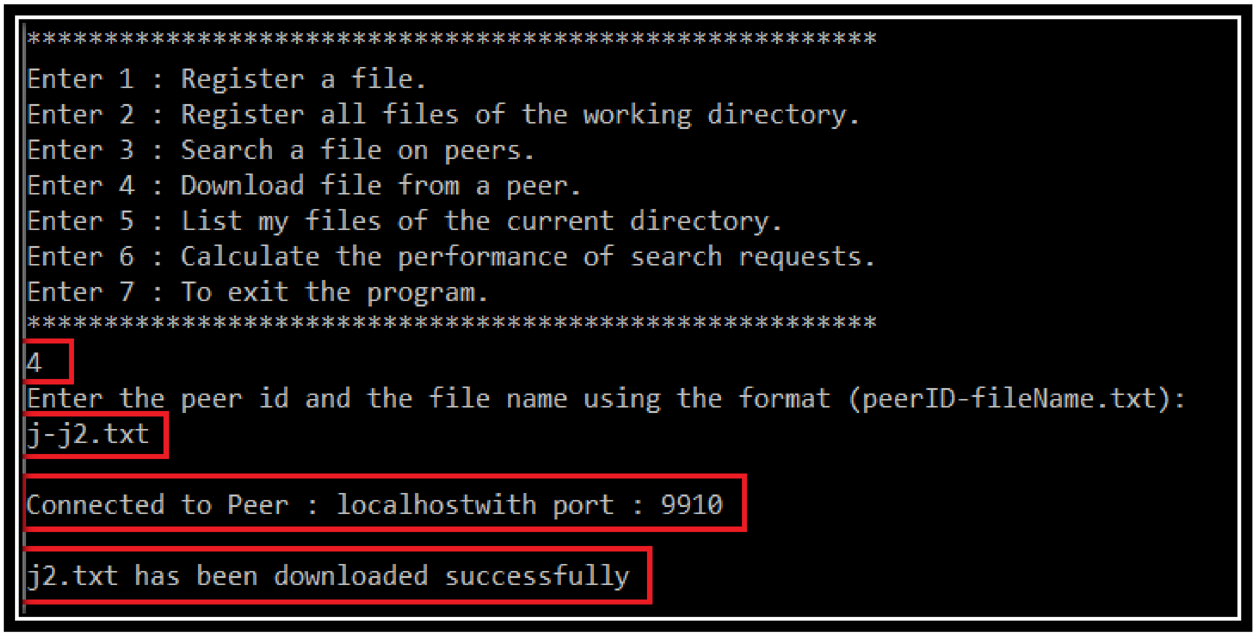
- And Peer J will have a message like this on its screen. But with the file name j2.txt.



```
File a2.txt has been found on this peer !

Hit query has been sent back to the main searcher !
```

4. Download a file from a Peer.

Now as we know that file j2.txt is at Peer j. We can directly download it by choosing the 4th option. After selecting the 4th option the user has to provide with the peer name and the filename in the given format. For e.g. Suppose user wants to download file "**j2.txt**" from **Peer J** then the input should be in format "**j-j2.txt**".



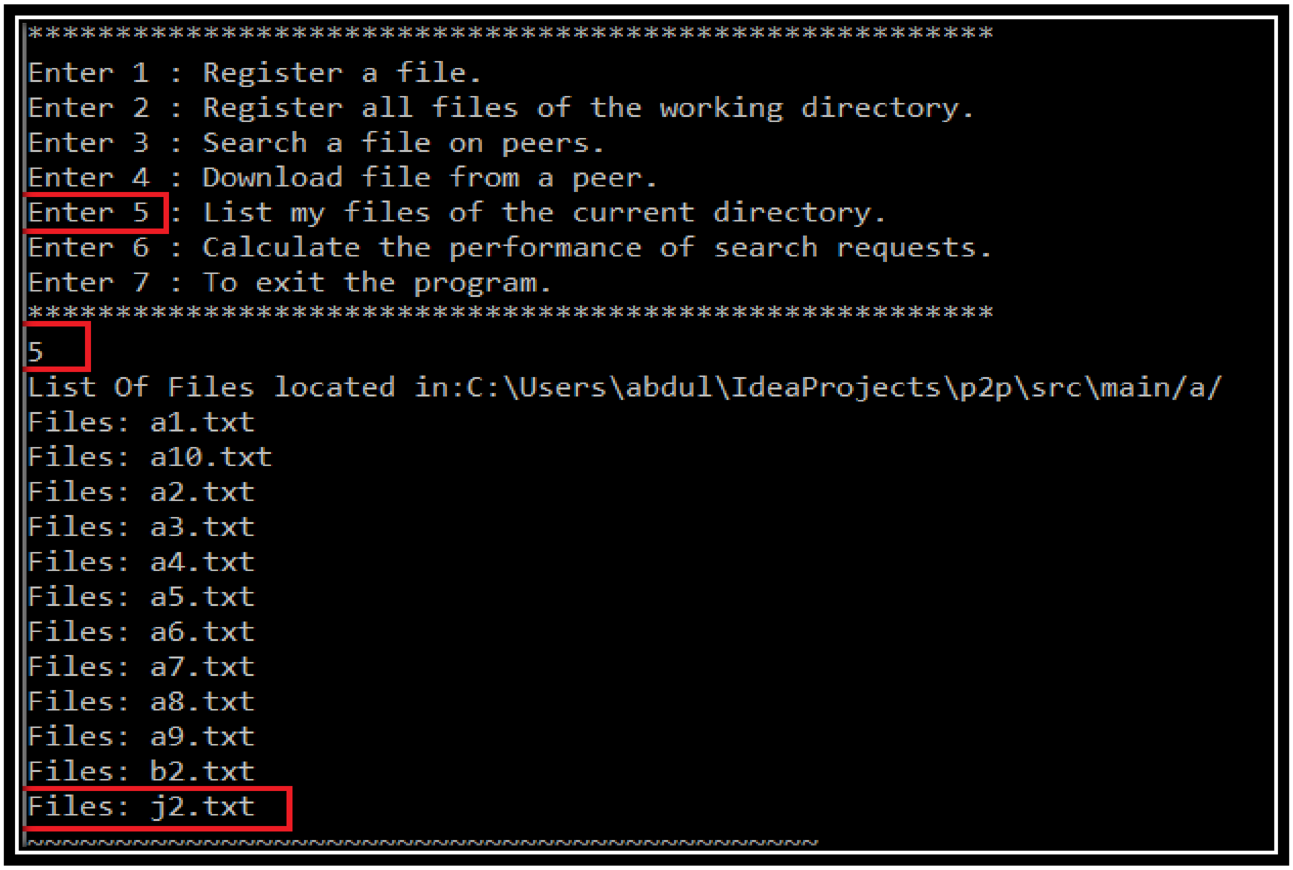
```
*******************************************************
Enter 1 : Register a file.
Enter 2 : Register all files of the working directory.
Enter 3 : Search a file on peers.
Enter 4 : Download file from a peer.
Enter 5 : List my files of the current directory.
Enter 6 : Calculate the performance of search requests.
Enter 7 : To exit the program.
*******************************************************
4
Enter the peer id and the file name using the format (peerID-fileName.txt):
j-j2.txt

Connected to Peer : localhostwith port : 9910

j2.txt has been downloaded successfully
```

Figure 21. Downloading a file from a Peer

After this step the file j2.txt would be transferred to the directory of Peer a, which we can see by choosing the 5th option of the main menu.

5. List the files of the current directory.

To view all the files on a Peer, user has to choose the option 5 on the main menu. The console output will show the path of the Peer files and the list of all the files on the Peers. Also, the files downloaded from other Peers.

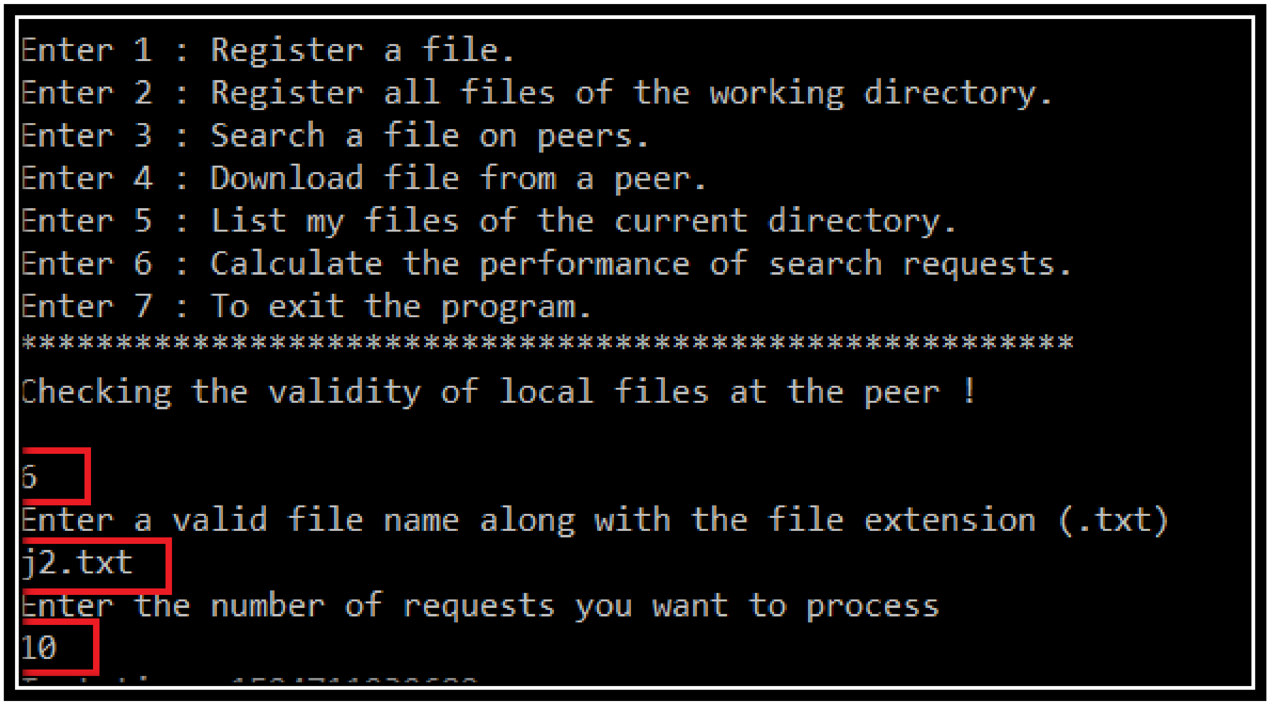


```
*******************************************************
Enter 1 : Register a file.
Enter 2 : Register all files of the working directory.
Enter 3 : Search a file on peers.
Enter 4 : Download file from a peer.
Enter 5 : List my files of the current directory.
Enter 6 : Calculate the performance of search requests.
Enter 7 : To exit the program.
*******************************************************
5
List Of Files located in:C:\Users\abdul\IdeaProjects\p2p\src\main/a/
Files: a1.txt
Files: a10.txt
Files: a2.txt
Files: a3.txt
Files: a4.txt
Files: a5.txt
Files: a6.txt
Files: a7.txt
Files: a8.txt
Files: a9.txt
Files: b2.txt
Files: j2.txt
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Figure 22. List of files on Peer "a"

Now, we can notice in the above screenshot that the file “**j2.txt**” has been transferred to the directory of Peer A as it is being listed in the name of the files **Peer a** owns.

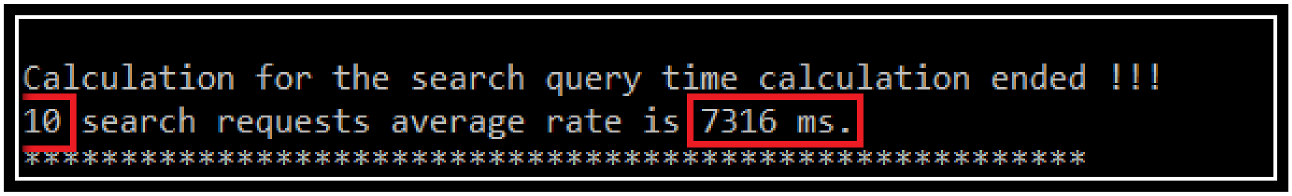
6. Calculation of Performance of search request.



```
Enter 1 : Register a file.
Enter 2 : Register all files of the working directory.
Enter 3 : Search a file on peers.
Enter 4 : Download file from a peer.
Enter 5 : List my files of the current directory.
Enter 6 : Calculate the performance of search requests.
Enter 7 : To exit the program.
*******************************************************
Checking the validity of local files at the peer !

6
Enter a valid file name along with the file extension (.txt)
j2.txt
Enter the number of requests you want to process
10
```

Figure 23. Calculation of Performance of search request

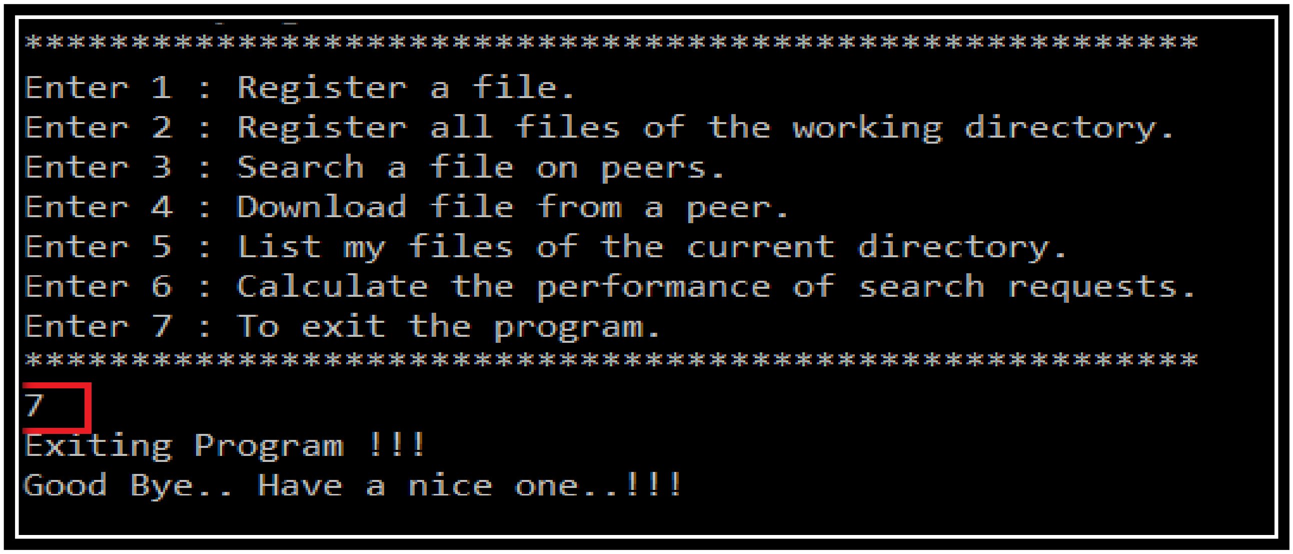
The above result of the calculation will give us the average time taken to make 10 search requests.



```

Calculation for the search query time calculation ended !!!
10 search requests average rate is 7316 ms.
*******************************************************
```

Figure 24. Calculation Done for 10 search requests.

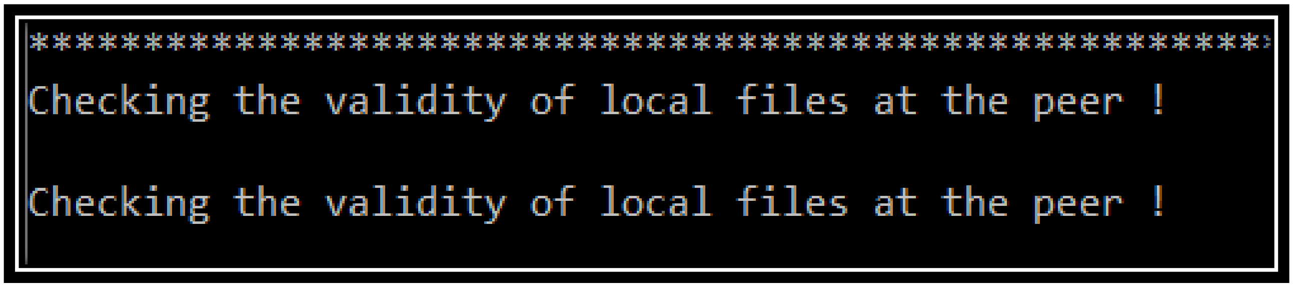
7. To exit the Program



```
*******************************************************
Enter 1 : Register a file.
Enter 2 : Register all files of the working directory.
Enter 3 : Search a file on peers.
Enter 4 : Download file from a peer.
Enter 5 : List my files of the current directory.
Enter 6 : Calculate the performance of search requests.
Enter 7 : To exit the program.
*******************************************************
7
Exiting Program !!!
Good Bye.. Have a nice one..!!!
```

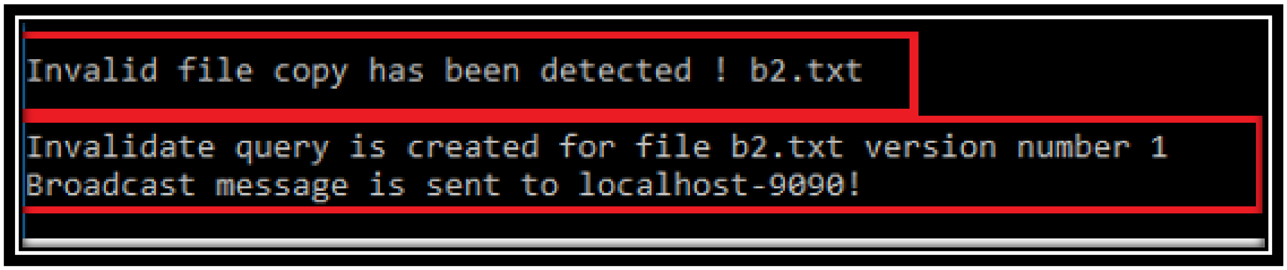
Figure 25. Exiting the Program

A thread will be running in the background for all the peers to check the validity of the files.



```
********************************************************

Checking the validity of local files at the peer !

Checking the validity of local files at the peer !
```

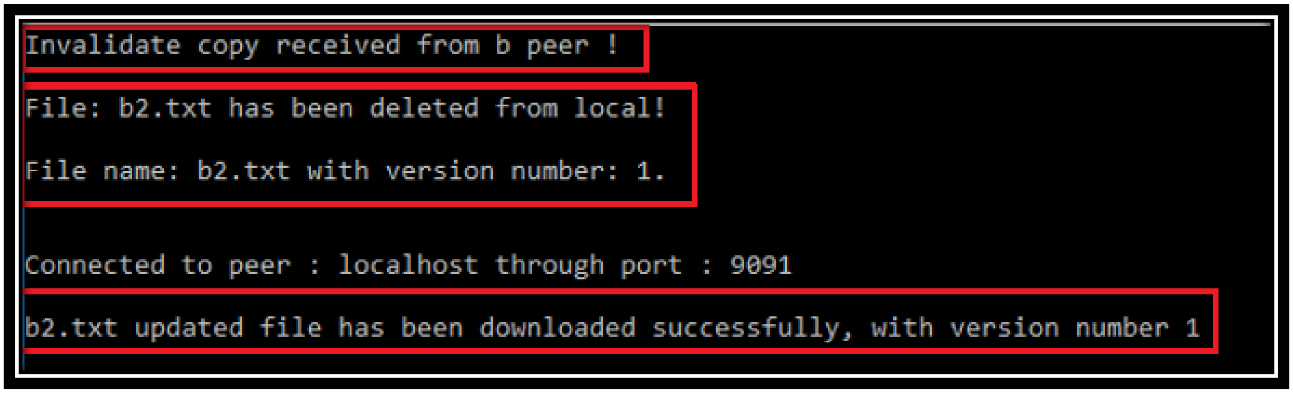
If there will be any change in the files it will give an error message. For eg. “**b2.txt**”.



```
Invalid file copy has been detected ! b2.txt
Invalidate query is created for file b2.txt version number 1
Broadcast message is sent to localhost-9090!
```

Figure 26. Invalid file detection

The Peer downloading the file will receive this message on its console.

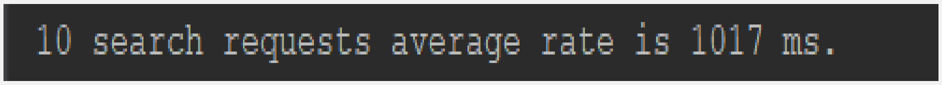
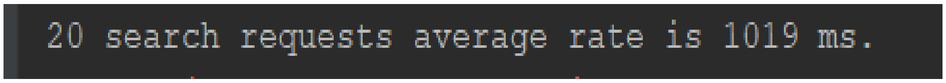
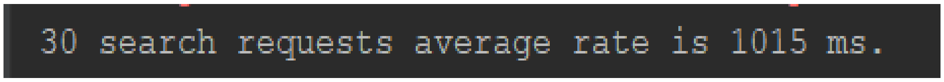


```
Invalidate copy received from b peer !
File: b2.txt has been deleted from local!

File name: b2.txt with version number: 1.

Connected to peer : localhost through port : 9091
b2.txt updated file has been downloaded successfully, with version number 1
```

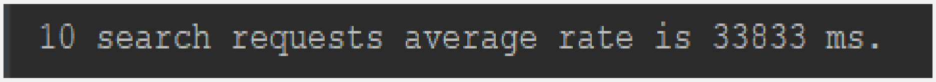
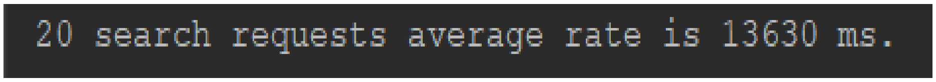
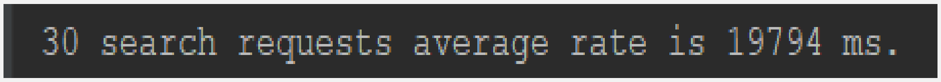
Figure 27. Invalid file copy deletion

E. Table of Results for search query calculation

STAR TOPOLOGY

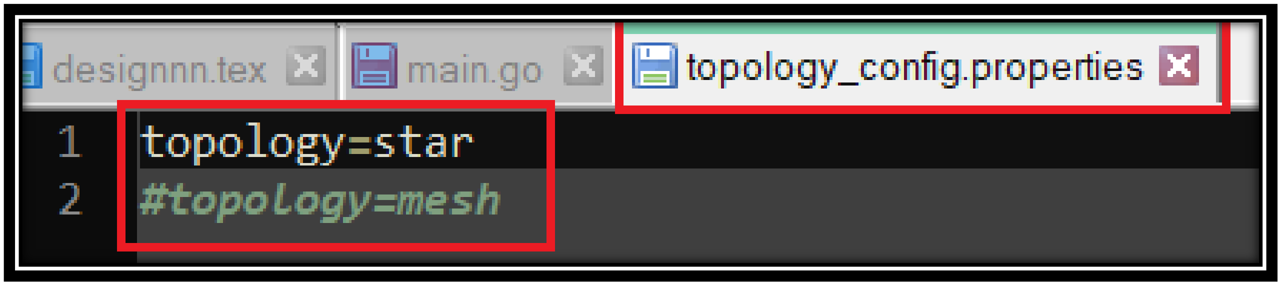
10	 `10 search requests average rate is 1017 ms.`
20	 `20 search requests average rate is 1019 ms.`
30	 `30 search requests average rate is 1015 ms.`

MESH TOPOLOGY

10	 `10 search requests average rate is 33833 ms.`
20	 `20 search requests average rate is 13630 ms.`
30	 `30 search requests average rate is 19794 ms.`

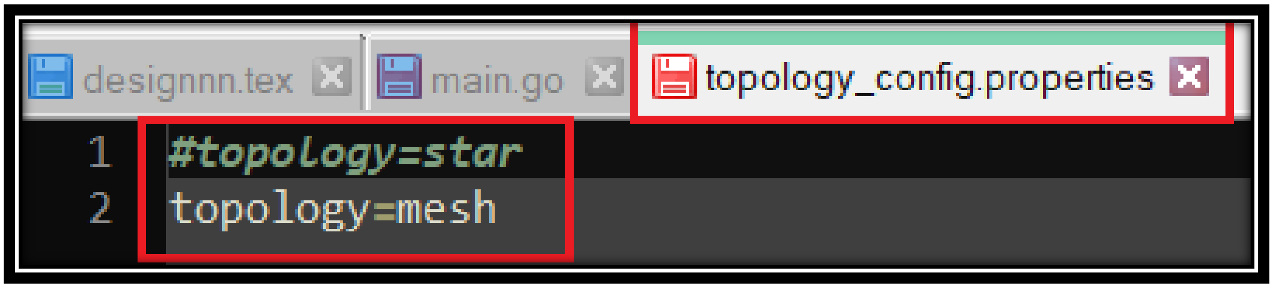
F. Topology Information:

The Program has been built for two different topologies that are “**mesh**” and “star” topology. By default, the program will have the topology to be set as **star**. If the user wants to run the mesh topology, user has to open the topology configuration file named “topology_config.properties”. which looks like the image below. And comment the line “topology=star” and uncomment “topology=mesh”



```
topology=star
#topology=mesh
```

Figure 28. Star Topology



```
#topology=star
topology=mesh
```

Figure 27. Mesh Topology

G. Files Owned by Peers

Peer A	Peer B	Peer C	Peer D	Peer E	Peer F	Peer G	Peer H	Peer I	Peer J
a1.txt	b1.txt	c1.txt	d1.txt	e1.txt	f1.txt	g1.txt	h1.txt	i1.txt	j1.txt
a2.txt	b2.txt	c2.txt	d2.txt	e2.txt	f2.txt	g2.txt	h2.txt	i2.txt	j2.txt
a3.txt	b3.txt	c3.txt	d3.txt	e3.txt	f3.txt	g3.txt	h3.txt	i3.txt	j3.txt
a4.txt	b4.txt	c4.txt	d4.txt	e4.txt	f4.txt	g4.txt	h4.txt	i4.txt	j4.txt
a5.txt	b5.txt	c5.txt	d5.txt	e5.txt	f5.txt	g5.txt	h5.txt	i5.txt	j5.txt
a6.txt	b6.txt	c6.txt	d6.txt	e6.txt	f6.txt	g6.txt	h6.txt	i6.txt	j6.txt
a7.txt	b7.txt	c7.txt	d7.txt	e7.txt	f7.txt	g7.txt	h7.txt	i7.txt	j7.txt
a8.txt	b8.txt	c8.txt	d8.txt	e8.txt	f8.txt	g8.txt	h8.txt	i8.txt	j8.txt
a9.txt	b9.txt	c9.txt	d9.txt	e9.txt	f9.txt	g9.txt	h9.txt	i9.txt	j9.txt
a10.txt	b10.txt	c10.txt	d10.txt	e10.txt	f10.txt	g10.txt	h10.txt	i10.txt	j10.txt

End of Test Document