

Testing Verification Document

Following Test cases were run against the system which worked correctly.

Peer Test cases:

Sr. No.	Description	Input	Expected O/p	Actual o/p	Result
1.	Normal lookup and download of file (present in only 1 peer)	./run_peer.sh	The index server should return only 1 host (peer). After user select that host, the file should get downloaded in its shared directory.	The Index server shows IP of only peer 1. After user select that IP, the file id downloaded on peer 2.	PASS
2.	Normal lookup and download of file (present in 2 or more peers)	./run_peer.sh	The index server should return the hosts (peers). After user select any host, the file should get downloaded from that peer to peer1's shared directory.	The index server prints all the hosts where file is present. After user selects any host, the file gets downloaded from that peer to peer1's shared directory.	PASS
3.	Lookup for file not present in any of the peers (hence not indexed in the	./run_peer.sh	The index server should return 0 results for the hosts (peers) where file is available and	Index server returns 0 results (peers) and peer prints error message " <i>No peer found</i> "	PASS

	indexed server)		the peer should print appropriate message.	<i>for the file named abc.txt. Please try again!"</i>	
4.	User enters an incorrect option (number) while asked for the host number to select.	./run_peer.sh	The peer should show an error message and ask to select appropriate option.	The peer shows an error message <i>"Incorrect choice. Please try again."</i>	PASS
5.	Try to look up and download multiple files of varying sizes from another peer.	./run_peer.sh	All files should get downloaded and the time taken should increase linearly.	All files get downloaded and the time increases linearly. Please see performance testing doc for details.	PASS
6.	Send 1000 consecutive request (using 1 peer). File getting downloaded from another peer.	./run_peer_performance.sh	All the 1000 requests should be successful and the files should get downloaded.	All 1000 requests are successful and the files get downloaded. The total time taken is also displayed.	PASS
7.	Send 1000 consecutive request (using 2 peer). Both peers are downloading files from each other and sending simultaneous	./run_peer_performance.sh	All the 1000 requests on both peers should be successful and the files should get downloaded.	All 1000 requests on both peers are successful and the files get downloaded. The total time taken is	PASS

	requests to index server.			also displayed.	
8.	Send 1000 consecutive request (using 3 peers). All 3 peers downloading from each other and sending simultaneous requests to index server.	./run_peer_performance.sh	All the 1000 requests on all 3 peers should be successful and the files should get downloaded.	All 1000 requests on all 3 peers are successful and the files get downloaded. The total time taken is also displayed.	PASS

Directory Scanner Test Cases:

Sr. No.	Description	Input	Expected O/p	Actual o/p	Result
9.	All the file in shared directory should be registered with index server on startup of the peer.	./run_peer.sh	The files present in the shared directory should get indexed at the indexed server on startup.	The files present in the shared directory gets indexed at the indexed server on startup, with the peer id of the peer.	PASS
10.	If a file is added to the shared directory while peer is running, it should be detected and registered on index server.	./run_peer.sh	When a file is added to the shared folder, it should automatically get registered with index server under the corresponding peer id.	The newly added file gets indexed at the index server. The directory is scanned every 10 seconds and the changes are reflected.	PASS

11.	If a file is deleted from the shared directory while peer is running, it should be detected and unregistered on index server.	./run_peer.sh	When a file is deleted from the shared folder, it should automatically get un-registered with index server from the corresponding peer id.	The deleted file's entry gets removed the index server. The directory is scanned every 10 seconds and the changes are reflected.	PASS
-----	---	---------------	--	--	------

Index Server Test Cases:

Sr. No.	Description	Input	Expected O/p	Actual o/p	Result
12.	Allow multiple requests to be handled simultaneously. 3 peers were sending requests to the index server concurrently.	./run_index_server.sh	The index server should handle all the simultaneous concurrent request and return the correct results to the peers.	While sending 1000 continuous requests from 3 concurrent servers, all the requests were serviced properly.	PASS
10.	A file could be present in more than one peers. The indexing server should be able to handle it.	./run_index_server.sh	When the same file is present on multiple peers, the index server should be able to handle it properly and keep record on all	The index server returns multiple hosts to the peer in case of same file being present in multiple peers.	PASS

			peers with the file.		
11.	When a file is not present in any peer, the index server should return 0 results.	./run_peer.sh	When a file is not present on any peer, the index server should return 0 results, and appropriate message should be displayed.	When a file is not present on any peer, the index server returns 0 results, and appropriate message is displayed.	PASS