The system has three main modules : **UserHandler**, **SDFSMessageHandler** and **FailureHandler**.

1. **UserHandler**: This module react to the commands executed by users. By interacting with other processes in the group, the module completes functions such as **Put**, **Get**, **Delete, Ls** and **Store**.
2. **SDFSMessageHandler**: This module is the centre of message exchange. It listens for different types of messages and make responds as shown in the diagram below. When a message is received from a remote process, a new thread will be spawned to handle the specific request. Messages is processed in a parallelized manner. No bottleneck will effect the overall performance of the system.
3. **FailureHandler**: This module is to make reaction to nodes' failures. The module has two sub-modules. One is **MasterElection.** When a master process fails, master election procedure is triggered within the group. The detail of master election is described in diagram below. The other is **ReplicaRenewal.** When a process fails, the master will make new replicas for all files used to store in that process.

Bootstrapping: Whenever a new process is brought up from a cold place, it will clean the old stage by making a new SDFS file folder. User should specify the leader when start a process. The new joined process will contact with the leader and get the member list as well as the master list. If the process is specified to be a master, it will also request for a file log from leader. The leader will broadcast all process about the new joined member.

Leader: The leader use the same assumption as in the MP2. It manages the join procedure of a new process. The leader should be one of the master in this SDFS system.

Master: The distributed system always has 3 masters in the same time, so the system can tolerate 2 simultaneous failures while maintaining the correct function. The master keeps a file log, which store the information of places of file replicas. The response for all file operations as it will decide where to write the file replica and tell other nodes the exact places of a specific file. When one or two master fails, the masters alive will call for election procedure by choosing one or two processes which as a biggest timestamp. Once the new masters are elected, the old masters will broadcast to all processes.

Replica Strategy: When user execute a **put** command, the **UserHandler** will contact one of the masters by sending a **putwhere** message. The master will return a list with 3 processes. Once received the list from master, the process will send **write** message to all 3 processes in the list while sending them the file. The process receiving **write** message will then accepts the file and store it into the SDFS file system. Once a process done with writing file, it will send a **listEntry** message to masters. The masters will be informed that a specific file is written in a node and add that information to the file log. If a process want to get a SDFS file that does not exist in the process, it will send a **getwhere** message to the master and get a list of processes that has the file. Then it will send a **read** message to a place that has the file. The replica place will then respond with the requested file. When a process fails, one of the master will start the **replica renewal** procedure by checking the files used to exist in the failing process. For each file, the master will inform a current owner of that file by sending a **renew** message with a new place

for replica. Once received a **renew** message, the process will send **write** message with the file to the new place. When the user put the **Delete** command, the process will send a **delete** message to one of the masters. The master will forward the **delete** message to the process which contains the file and told them to delete the file from SDFS file system.

Utilizing former MPs: The distributed grep developed in **MP1** came in handy when taking the logs from various processes and trying to figure out where the messages were being lost or when the appropriate response was not being sent back. The failure detector in **MP2** is used to keep a member list in each process as well as inform the SDFSMessageHandler about the failures.

Bandwidth Usage for re-replica:

Message type	Bandwidth
write+ filename	18 B
Renew + filename	40 B
Actual file	30 MB
listEntry + fileplace+filename	59 B
Clear + filename	18 B
Other communication cost	15 B

Total bandwith is 30 MB/0.62s = 48.4 MB/s

Working Efficiency:

1. We do write and read operation for both 500MB and 20MB file. The time measurements is shown in the following graph:

File size	Average writing time	Average reading time
500 MB	3.5 s	2.4s
20 MB	0.4 s	0.2 s

2. We also measured the replica renewal time when there was failures: The time to detect failure and replicate 30 MB file take average 0.62 seconds.
3. We measured the time storing a 1.3 GB file from wikipidia on 4 machines and 8 machines. The statics is shown in the graph:

Nodes amount	Average time cost
4	7.5 s
8	9.2 s