ELEVATOR OS

We have decided to develop an elevator system. An elevator operating system is under the real time operating system type. Real time operating systems make use of the available devices, thus have the maximum utilization.
The Java programming language has been selected for implementation so that an application should be executed in the form of an applet.

The scheduling that we followed in our Operating System is **fixed priority scheduling.**
We used Queues and Arraylists.
The memory contains all the processes, which are:
-***Screen Time Process***, which is responsible for viewing the current time on a screen in the elevator.
-***Screen Current Floor Process***, which views the floor that the elevator is currently at.
-***Fan Process***, which is responsible for turning on/off the fan. It turns on when the elevator is moving and turns off when it's not moving.
-**Move Process,**
-***OpenDoor Process*****,** which opens the door, the door opens when the elevator is not moving.
-*CloseDoor Process***,** which closes the door, the door must be closed when the elevator is moving.
-A queue of ***High Priority Processes*****,** which chooses which processes to execute first.
-A queue of ***Medium Priority Processes*****,** which chooses which processes to execute between high and low priority processes.
-An ArrayList of ***Low Priority Processes*****,** which chooses which processes to execute last.
-An ArrayList of ***No Priority Processes*****,** which contains the processes that can be executed at any time.

-The memory class also contains an integer named ***runningProcesses*** which is initially set to zero and keeps track of how many processes are currently being executed.
-An integer named ***currentFloor*** which saves the value of the floor that the elevator is currently at.
-An integer named ***maxFloor*** which saves the value of maximum number of floors the elevator can reach.
-An integer named ***minFloor*** which saves the value of minimum number of floors the elevator can reach.
-Our elevator contains 4 floors. Floor 0 to Floor 4.
-A queue of **destinationFloors,** which contains the desired floor number that the user wants to go to.

For our applications we implemented:
-**A Fan,** A fan thread that starts whenever the elevator is moving and stops whenever the elevator stops

-**Move up,** A thread that starts when an elevator is moving up and stops whenever the elevator stops.
-Move down,A thread that starts when an elevator is moving up and stops whenever the elevator stops.
-Open door, A thread that starts when an elevator has arrived to its destination and closes after a certain amount of time
-Close door, A thread that starts when the openDoor thread closes and end when the openDoor thread starts