

1-Modular Priority Engine :

File	Function	Role
Delivery.h	calculatePriorityScore()	Performs modular score calculation
ConfigurationManager.h/.cpp	getWeight(), getServiceTypeScore()	Provides weight and score values
DeliveryManager.cpp	addDelivery()	Calls calculatePriorityScore() before enqueueing
AdminConsole.cpp	addDelivery()	Accepts user input

Result :

```
9. Exit
Enter your choice: 6
Enter Delivery ID: 20
Enter Destination: test
Enter Delivery Type (0=URGENT, 1=STANDARD, 2=FRAGILE): 0
Enter Estimated Time (minutes): 21
Added urgent delivery: 20
--- Queued Deliveries with Scores ---
Urgent (1 deliveries):
ID: 20, Score: 4.5
Standard (0 deliveries):
Fragile (0 deliveries):
Delivery added successfully.
```

2. Dynamic Priority Updates:

Component	Function(s)	File(s)
Track waiting time	entryTime, getWaitingTime()	Delivery.h
Score recalculation	calculatePriorityScore()	Delivery.h
Starvation avoidance	boostPriority()	Delivery.h
Queue update logic	updatePriorities()	DeliveryManager.cpp
External trigger	applyFairnessBoost()	DeliveryManager.cpp
Heap structure	enqueue()	PriorityQueue.h, MaxHeap.h

Result :

```
--- Smart Queue Management System ---
1. Configure System Parameters
2. Run Simulation
3. View Stats
4. Modify Queue Policies
5. Generate Report
6. Add Delivery
7. Cancel Delivery
8. View Cancelled Deliveries Log
9. Exit
Enter your choice: 1
==== Configure System Parameters ====
Enter urgency weight (current: 0.5): 0.5
Enter waiting time weight (current: 0.3): 0.4
Enter service type weight (current: 0.2): 0.5
Enter max wait time for fairness boost (current: 25): _
```

3. Multiple Queues with Merging:

Functionality	Function Name / Concept	File Implemented In
Declare and manage separate queues	urgentDeliveries, standardDeliveries, fragileDeliveries	DeliveryManager.h
Add delivery to the appropriate queue	addDelivery()	DeliveryManager.cpp
Process deliveries in priority order	processNextDelivery()	DeliveryManager.cpp
Merge STANDARD into URGENT if URGENT is empty	mergeQueues()	DeliveryManager.cpp
Merge FRAGILE into URGENT if still empty	mergeQueues()	DeliveryManager.cpp
Queue structure (max-priority)	PriorityQueue<T> class	PriorityQueue.h
Heap behavior used in queues	MaxHeap<T> implementation	MaxHeap.h

4. Fairness Monitor:

Functionality	Function Name	File Implemented In
Check how long a delivery has been waiting	getWaitingTime()	Delivery.h
Compare with fairness threshold & boost if exceeded	boostPriority()	Delivery.h
Apply fairness to all queued deliveries	updatePriorities()	DeliveryManager.cpp
Entry point for fairness monitor logic	applyFairnessBoost()	DeliveryManager.cpp
Retrieve max wait threshold	getMaxWaitTime()	ConfigurationManager.cpp
Retrieve boost multiplier value	getBoostMultiplier()	ConfigurationManager.cpp
Set fairness-related values (admin interface)	modifyQueuePolicies()	AdminConsole.cpp

Result :

```
Enter max wait time for fairness boost (current: 25): 30
Enter fairness boost multiplier (current: 0.5): 0.5
Enter simulation duration (minutes) (current: 60): 60
Enter simulation arrival rate (individuals/minute) (current: 0.5): 0.5
Enter number of service counters (current: 3): 3
Enter URGENT service type score (current: 10): 10
Enter STANDARD service type score (current: 5): 5
Enter FRAGILE service type score (current: 8): 8
Configuration updated successfully.

--- Smart Queue Management System ---
1. Configure System Parameters
2. Run Simulation
3. View Stats
4. Modify Queue Policies
5. Generate Report
6. Add Delivery
7. Cancel Delivery
8. View Cancelled Deliveries Log
9. Exit
Enter your choice: 6
Enter Delivery ID: wee
Enter Destination: weeeee
Enter Delivery Type (0=URGENT, 1=STANDARD, 2=FRAGILE): 0
Enter Estimated Time (minutes): 21
Added urgent delivery: wee
--- Queued Deliveries with Scores ---
Urgent (1 deliveries):
ID: wee, Score: 7.5
```

5. Simulation Mode:

Functionality	Function Name	File Implemented In	Expected Output / Behavior
Run the entire delivery processing simulation	runSimulation()	SimulationManager.cpp	Begins simulation loop; processes deliveries by priority; simulates timing.
Handle one delivery at a time from queues	processNextDelivery()	DeliveryManager.cpp	Dequeues highest priority delivery and logs its service start/end time.
Apply fairness boosts during simulation	applyFairnessBoost()	DeliveryManager.cpp	Calls updatePriorities() to fairly adjust scores of long-waiting deliveries.
Merge queues if urgent is empty	mergeQueues()	DeliveryManager.cpp	Moves deliveries from other queues to urgent if needed.
Get number of processed deliveries	getProcessedDeliveriesCount()	SimulationManager.cpp	Returns count of deliveries served during the simulation.
Get total queue size	getQueueSize()	SimulationManager.cpp	Shows how many deliveries are still waiting during or after simulation.
Start the simulation (admin side)	start() (case 2 in Admin Console)	AdminConsole.cpp	User chooses option 2. Run Simulation; output simulates the process step by step.

Result :

```
5. EXIT
Enter your choice: 2
Starting simulation for 60 minutes with arrival rate 0.5 and 3 counters.

--- Time: 0 minutes ---
Urgent Queue Size: 0
Standard Queue Size: 0
Fragile Queue Size: 0

--- Time: 1 minutes ---
Urgent Queue Size: 0
Standard Queue Size: 0
Fragile Queue Size: 0

--- Time: 2 minutes ---
Urgent Queue Size: 0
Standard Queue Size: 0
Fragile Queue Size: 0

--- Time: 3 minutes ---
Added standard delivery: D1254
New Arrival: ID=D1254 (P=2.5)
Processed: ID=D1254 (P=2.5)
```

6. Advanced Reporting and Sorting:

Functionality	Function Name / Logic	File Implemented In	Output / Effect
Generate reports filtered by delivery type	generateReport(filter, sort)	ReportManager.cpp	Shows only deliveries of type urgent, standard, or fragile if filter is provided
Sort reports by priority or waiting time	generateReport() logic	ReportManager.cpp	Sorts by priorityScore by default, or waitingTime if specified by admin
Provide admin input for filter/sort options	generateReport()	AdminConsole.cpp	Admin asked: "Enter filter type" and "Enter sort criteria" via std::getline()
Format and print the report	print() on Delivery object	Delivery.h	Displays Delivery ID, Type, Est. Time, and Priority Score in readable format

Results:

```
delivery_report.csv X
delivery_report.csv
1 ID,Type,Priority,Wait Time,Service Time
2 D1168,urgent,4.50,0.00,0.08
3 D687,urgent,4.50,0.00,0.23
4 D8969,urgent,4.50,0.00,0.18
5 D4074,urgent,4.50,0.00,0.13
6 D3227,urgent,4.50,0.00,0.23
7 D1457,urgent,4.50,0.00,0.15
8 D296,urgent,4.50,0.00,0.15
9 D5335,urgent,4.50,0.00,0.13
10 D8326,urgent,4.50,0.00,0.13
11 D5512,urgent,4.50,0.00,0.23
12 D8797,urgent,4.50,0.00,0.12
13 1,urgent,4.50,0.22,0.08
```

7. Admin Console:

Functionality	Function Name	File(s)	Output / Behavior
Display main menu	start()	AdminConsole.cpp	Shows numbered options (Add, Run, Report, Exit, etc.) in the console
Add new delivery	addDelivery()	AdminConsole.cpp	Prompts for ID, destination, type, est. time; then adds to queue
Cancel a delivery by ID	cancelDelivery()	AdminConsole.cpp	Prompts for ID, removes it from active queue if found, adds to cancel log
View cancelled delivery log	viewCancelledDeliveries()	DeliveryManager.cpp	Lists all cancelled deliveries (most recent first)
View live queue stats	viewStats()	AdminConsole.cpp	Shows queue sizes and processed count
Modify system settings	modifyQueuePolicies()	AdminConsole.cpp	Allows changing fairness settings and service counters via input
Run full simulation	runSimulation()	SimulationManager.cpp	Triggers processing, fairness, merging, output via simulation cycle

Generate filtered/sorted report	generateReport()	AdminConsole.cpp & ReportManager.cpp	Prompts for filter/sort; outputs report of processed deliveries
------------------------------------	------------------	---	---

Results:

```
--- Smart Queue Management System ---
1. Configure System Parameters
2. Run Simulation
3. View Stats
4. Modify Queue Policies
5. Generate Report
6. Add Delivery
7. Cancel Delivery
8. View Cancelled Deliveries Log
9. Exit
Enter your choice: 
```

8.Cancel Delivery “Bonus”:

Functionality	Function Name / Data Structure	File Implemented In	Output / Behavior
Cancel delivery by ID	cancelDelivery()	AdminConsole.cpp	Prompts user for ID; calls cancelDeliveryById(); shows success/failure message
Perform the actual cancellation	cancelDeliveryById()	DeliveryManager.cpp	Searches all queues; if found, removes delivery & pushes to cancelledStack
Stack data structure to log cancelled	std::stack<Delivery> cancelledStack	DeliveryManager.h	Stores deliveries in LIFO order for later logging
Track source of cancellation	Debug print logs (optional)	DeliveryManager.cpp	Shows [DEBUG] Checking... messages during cancellation search (if enabled)

Result :

```
--- Smart Queue Management System ---
1. Configure System Parameters
2. Run Simulation
3. View Stats
4. Modify Queue Policies
5. Generate Report
6. Add Delivery
7. Cancel Delivery
8. View Cancelled Deliveries Log
9. Exit
Enter your choice: 6
Enter Delivery ID: 2
Enter Destination: w
Enter Delivery Type (0=URGENT, 1=STANDARD, 2=FRAGILE): 2
Enter Estimated Time (minutes): 4
Added fragile delivery: 2
--- Queued Deliveries with Scores ---
Urgent (0 deliveries):
Standard (0 deliveries):
Fragile (1 deliveries):
ID: 2, Score: 3.6
Delivery added successfully.

--- Smart Queue Management System ---
1. Configure System Parameters
2. Run Simulation
3. View Stats
4. Modify Queue Policies
5. Generate Report
6. Add Delivery
7. Cancel Delivery
8. View Cancelled Deliveries Log
9. Exit
Enter your choice: 7
Enter Delivery ID to cancel: 2
Delivery 2 cancelled and logged.
```

9-View Cancelled Deliveries Log “Bonus”:

Functionality	Function Name / Logic	File Implemented In	Output / Behavior
View cancelled log	viewCancelledDeliveries()	DeliveryManager.cpp	Prints all cancelled deliveries from top of stack to bottom
Trigger view from menu	Menu case 8 in start()	AdminConsole.cpp	Admin selects "8. View Cancelled Deliveries Log" and this method is invoked
Display delivery info	print() method on Delivery	Delivery.h	Nicely prints ID, type, destination, time, score, etc. for each cancelled delivery

Results:

```
--- Smart Queue Management System ---
1. Configure System Parameters
2. Run Simulation
3. View Stats
4. Modify Queue Policies
5. Generate Report
6. Add Delivery
7. Cancel Delivery
8. View Cancelled Deliveries Log
9. Exit
Enter your choice: 8

--- Cancelled Deliveries Log (Most recent first) ---
Delivery ID: 2, Destination: w, Type: FRAGILE, Est. Time: 4 min, Priority Score: 3.6
```
