

## **TASK 2 - REPORT**

In this task, a queue data structure is used to manage treatment requests in a hospital system.

The queue follows the FIFO (First In First Out) rule.

This means the first patient who arrives is treated first.

Each treatment request is stored as a TreatmentRequest object.

The queue is implemented using a linked list, not a built-in Java queue.

The front of the queue stores the first request, and the rear stores the last request.

The enqueue method adds a new treatment request to the end of the queue.

This operation is done in  $O(1)$  time because the rear pointer is used.

The dequeue method removes the first request from the queue.

The front pointer moves to the next node.

This operation is also  $O(1)$ .

A queue is suitable for treatment requests because hospitals should treat patients in the order they arrive.

This structure is fair and easy to manage.

If a stack was used instead of a queue, the last patient would be treated first.

This is not correct for a hospital system.

Stack uses LIFO (Last In First Out) order, which can cause older patients to wait longer.

Both queue and stack operations have  $O(1)$  time complexity for add and remove.

However, the queue behavior is more suitable for real-life treatment systems.