1. **Analyze the time complexity of each operation (add, update, delete) in your chosen data structure.**

The time complexity for each operation:

1. Add Operation:

* Best Case: O(1)
* Average Case: O(1)
* Worst Case: O(n) (In case of hash collisions, which is rare if a good hash function is used)

1. Update Operation:

* Best Case: O(1)
* Average Case: O(1)
* Worst Case: O(n) (In case of hash collisions, which is rare if a good hash function is used)

1. Delete Operation:

* Best Case: O(1)
* Average Case: O(1)
* Worst Case: O(n) (In case of hash collisions, which is rare if a good hash function is used)

1. **Discuss how you can optimize these operations**

The current implementation of the inventory management system using a HashMap is already quite efficient, with O(1) average time complexity for add, update, and delete operations.