STL Containers

For this project, I need to create an inventory. The actual requirements of this are pretty vague though, as an inventory can work in a number of different ways. Is it a Pokemon style bag with no limit and different sections? Is it an inventory of a fixed size like in RPGs, etc? Or is it just a basic, bottomless bag?

I've opted for the last option there, purely for simplicity's sake. Using this system also narrows down my options, as per what can fit these requirements. I need something of theoretically unlimited size, dynamic and has features which allow for retrieval.

So far, I can rule out the following:

- Array: Static, not able to be resized.
- Deque: Don't really need queue functionality.
- Forward List: Not really the right choice here. Maybe good for a stack (If it weren't for stacks already existing.
- List: Getting closer, but still not really what I need. This is essentially just another queue.
- Set: Unique keys sorted by keys doesn't scream inventory to me.

I've left out anything passed this because they're just not suitable for many other reasons.

So what I'm left with is:

- Vector.
- Map.

The main thing here is that Vectors are supposed to be very efficient and very multipurpose. Maps, however are key-value pairs. Therefore, the key could be an item name, and the value could be the item data.

In this case, I'm planning on implementing the items as a class with its own name. Therefore, having a map, doubles up on the name. Though, it should be noted, I have not accounted for multiple items having an identical name. Considering the scale of this project, I doubt it will be necessary to have multiple items exist at once.

So, for now I've decided to go with a Vector (Though open to changing to a map later on depending on the requirements). This will mean that I'll have to implement my own search though. However, this isn't too hard to do, it'll just be a bit inefficient (maybe undoing the point of using a vector in the first place), but I'm now committed to it.