**Stock Management System**

**Project Requirement and Objective:**

**Objective: Create an application that helps user manage product stock**

**Functional Requirements:**

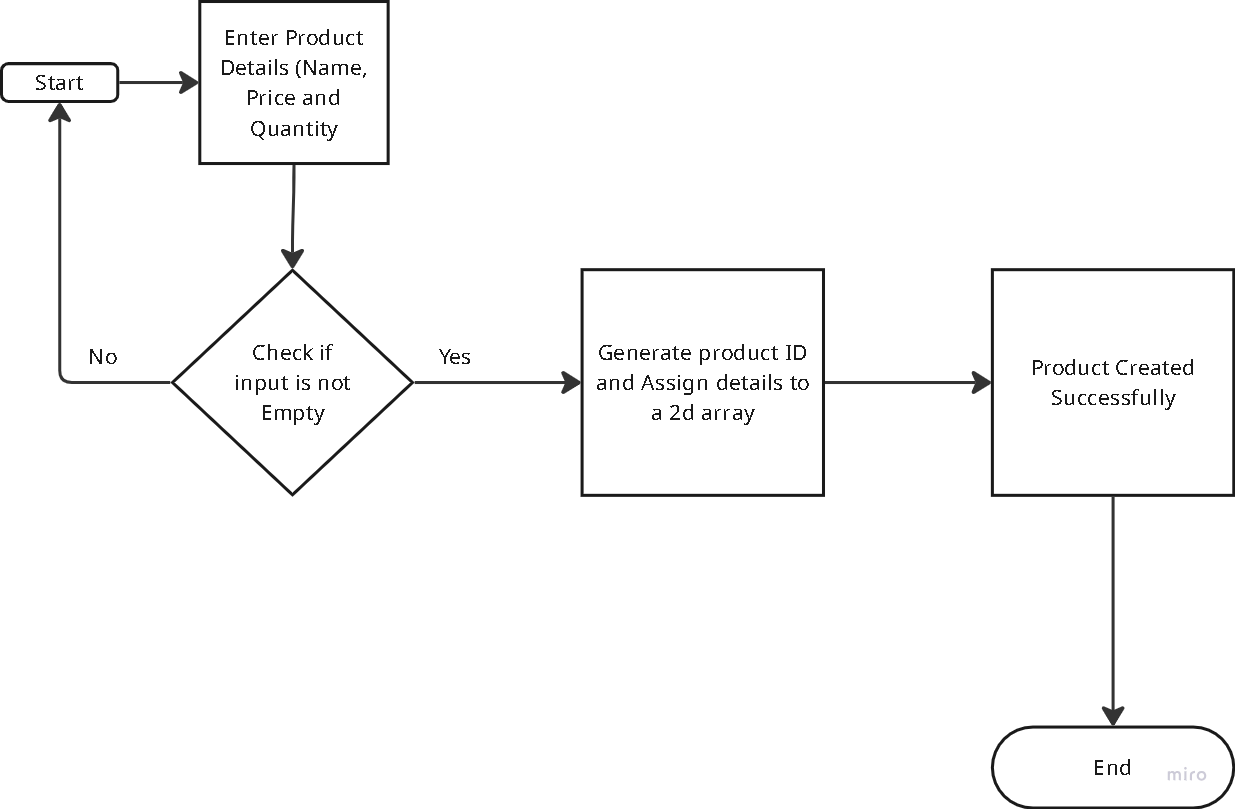
* Add new product (Id, Name, Quantity and Price)
* Update stock (When Sold or Restocked)
* View all products and stock level
* Remove product

**Non-Functional Requirements:**

* Menu Selection
* Readability
* Reusability
* Error handling

**Design Outline**

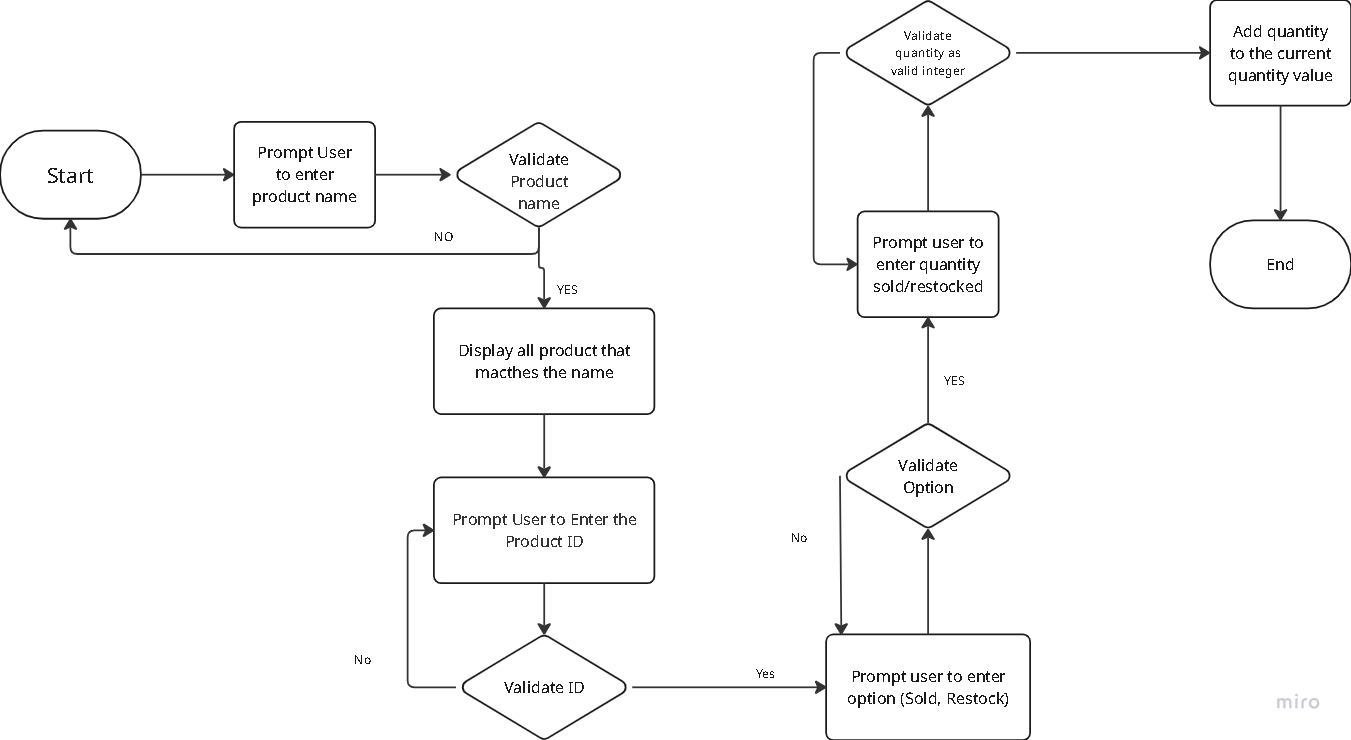
**Add Task:**



**Steps:**

* Prompt user to enter Name, Price and Quantity using Console.ReadLine()
* Check if input is not empty else return to step 1 using if/else statement
* Generate Product ID with name and product count using count variable in the for loop and concating name and count
* Assign the details to list using for loop
* Return true

**Update Stock:**



Steps:

* Prompt user for product name using Console.ReadLine()
* Validate product name using if/else statement
* Display all product that matches the name using for loop and if/else statement
* Prompt user for product ID using Console.ReadLine()
* Validate product ID using if/else statement
* Prompt user for option (sold/restock) using Console.ReadLine()
* Validate product option using if/else statement
* Prompt user for quantity using Console.ReadLine()
* Enusure quantity is numberic using if/else statement and TryParse()
* Update quantity using for loop and if/else statement
* Return true

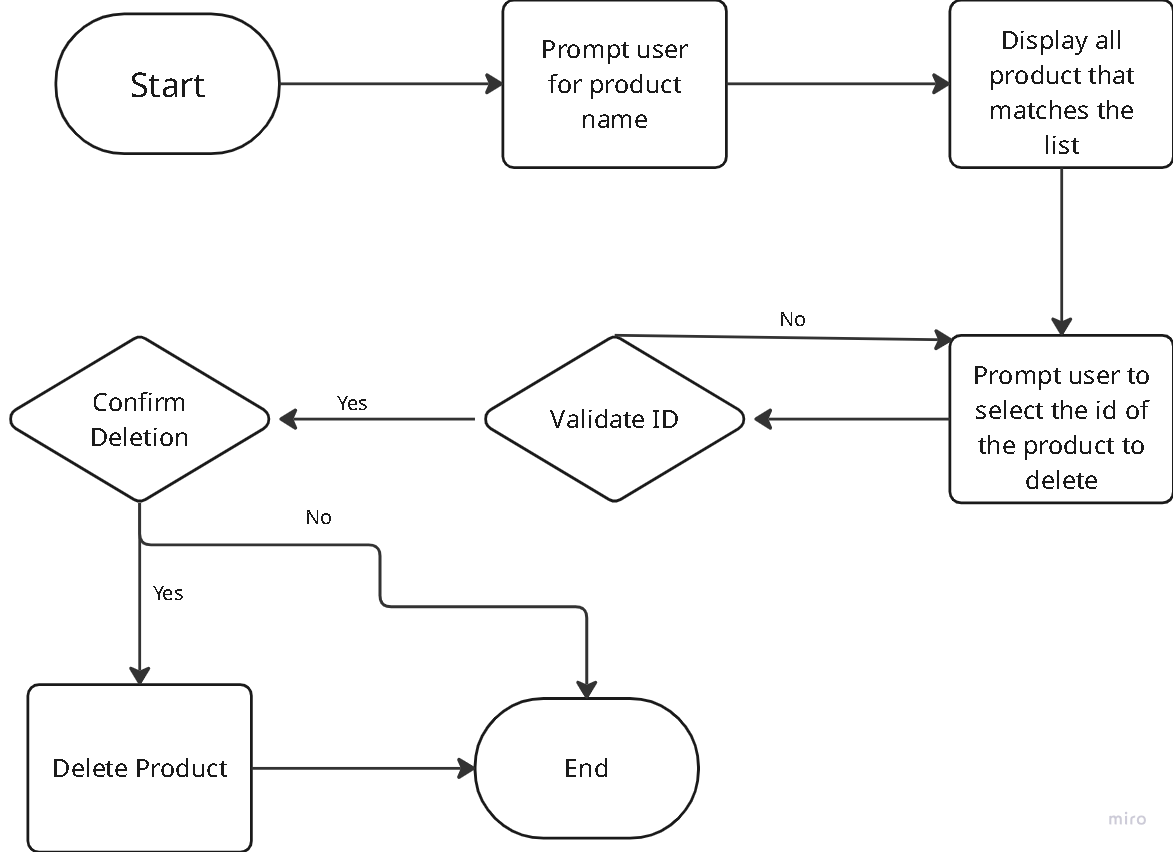
**View Products and Stock Level:**



**Steps:**

* Loop through product list using for loop
* Check if the list is empty
* Return false else display list
* Return true

**Remove Product**:



**Steps**:

* Prompt user to enter product name using Console.ReadLine()
* Display all product that matches the list using for loop and Console.WriteLine()
* Prompt user to enter product ID using Console.ReadLine
* Validate ID, using if/else. If yes,
* Confirm deletion using if/else. If yes
* Delete product