

INVENTORY

MANAGEMENT SYSTEM

# Department: Instrumentation & Control Engineering Branch: Cyber-Physical System

**Semester: III**

# Subject: Data Structures & Algorithms

**Submitted By:**

**Aryan Satpute**

# OVERVIEW OF INVENTORY MANAGEMENT SYSTEM

We have tried to implement an inventory management system using a binary search tree (BST) to store product information. It provides various functionalities, including adding products, displaying products, updating stock, recommending stock adjustments, recording sales, and viewing transaction history.

## Main Functionality:

Pre-adding Products: Initializes the inventory with a few pre-defined products.

User Interaction Loop: Presents a menu with options to add products, display products, update stock, recommend stock adjustments, record sales, view transaction history, or exit.

Product Addition: Prompts for product details and adds the product to the BST.

Product Display: Displays product information (name, SKU, stock, cost, cost price, units sold, revenue, profitability) sorted by SKU.

Stock Update: Prompts for SKU and quantity, updates stock in the BST, and handles underflow and stock alerts.

Stock Adjustment Recommendations: Traverses the BST and recommends stock adjustments based on revenue and sales.

Sales Recording: Traverses the BST, prompts for units sold for each product, updates stock and units sold, and handles stock exceedance. Transaction History View: Displays the transaction history (SKU and quantity) for all stock updates.

# BLOCK DIAGRAM

// Creating Products Include:

1. Name
2. SKU
3. Stock
4. Cost
5. Cost Price
6. Units Sold Calculate Revenue() Calculate Profitability ()

//Tracking Transaction Include

1. SKU
2. Quantity Transaction(SKU,Quantity)

//Collect Products Tree Node

{

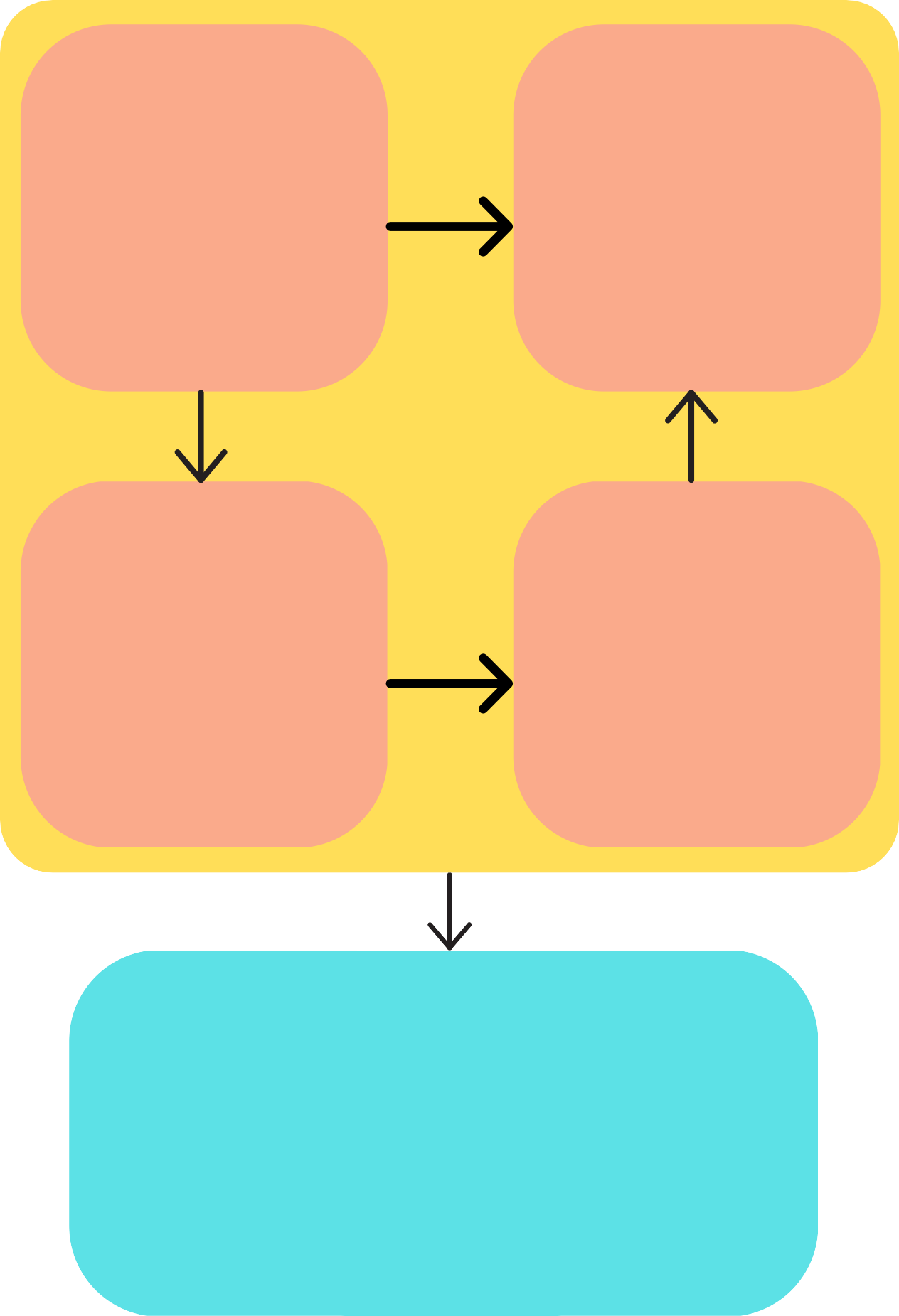
Product p; Left=Right=NULL;

};

//Including Functions to manage Inventory

InsertProduct() DisplayProduct() UpdateStock()

Recommend\_Stock\_Adjustment() Record\_Sale\_for\_Products() Get\_Transaction\_History()



//Create object of a class inventory management system InventoryManagementSystem ims;

Ims.preAddProduct();

//Create a menu with the following options:

* 1. Add Product
  2. Display Product
  3. Update Stock
  4. Recommend Stock Adjustment
  5. Record Sales
  6. View Transaction History
  7. Exit

# KEY COMPONENTS

Product: Represents a product with attributes like name, SKU, stock, cost, cost price, and units sold. The creation of new products takes place through this class.

Transaction: Represents a stock update transaction with SKU and quantity. This part is responsible for maintaining transaction records. This includes the addition of stocks, sales of stocks, updation of stocks, etc.

Tree Node: Represents a node in the BST, containing a Product object and pointers to left and right child nodes. We have used this data structure for maintaining records of products , for traversal and displaying of the stock information.

Inventory Management System: Manages the inventory, including adding products, updating stock, recording sales, and displaying various information. The aim of this part is to manage the inventory system.

main(): This is the driver part of the entire program. We can control the flow of the events, which would be executed using this part