ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY

UNIVERSITY FOR INDUSTRY

College of Electrical and Mechanical Engineering

Department of software engineering

Fundamental of programming II (SWEG2102)

Group -2

Project – Stock Management System

<u>Group members’ name</u>	<u>I’d number</u>
1.Yaiyneabeba Fuje	cep0110/14
2.Kalkidan Getnet	cep0065/14
3.Yohanes Tesfaye	cep0119/14
4.Hailemichael Yilma	cep0057/14

Submitted to:-Mr. Felix Edessa

Submission date :-Sep 17, 2023.

Contents **Page No.**

1.Introduction

We are proud introduce our cutting-edge stock management system designed specifically for businesses in Addis Ababa. The system is built using console-based application and using C++ and object-oriented principles to the adequate the special needs of the businesses in the region. The system offers automated recording and removal of products, and also avoids overstock and understocking moments .

We have tried to generate a practical system that is organized for local conditions, with features important features inventory tracking, stock level management, and automated scoring, generated using C++object-oriented programming, Other than creating user friendly environment, stock management system uses for preventing overstocking and understocking, to help businesses keep track of the location, helping businesses to manage their warehouse structure.

Stock management system is a valuable and essential tool for any type of business. As business progress to expand and develop their outcomes, the demand for stock management system becomes rising important. Our project states an advanced solution by generating a console based application in C++, using object- concepts.

2.Why stock management system for Addis Ababa?

Stock management system is essential for Addis Ababa, since it plays a significant role in the cities development and economic stability. Adequate stock management secures that the required amount of products is available at the time it is needed, to satisfy the demand of businesses and consumers. The system helps to prevent shortages, wastages, overstocking, understocking, reducing costs and up regarding income for businesses.

One of the main reasons why stock management is crucial for Addis Ababa is quick urbanization and population growth in the city. Since people move to the city for better job opportunities and higher standard living the demand for goods and services has outstandingly increased.

Moreover, Addis Ababa's position as the capital city of Ethiopia and even the continent Africa it's economic hub in addition spotlights the need for stock management system. The city is home to various industries, including manufacturing, hospitality and retail, which require a constant supply of goods and raw materials. Adequate stock management allows these industries to set up their functionalities, minimize supply chain distributions, and continual progress of goods, eventually contributing to development and economic stability. Efficient stock management system helps to minimize waste, reduce storage space requirements, and optimize resource utilization, there by driving sustainable practices in the city.

Generally stock management system is important for Addis Ababa, as it ensures a continual supply of goods to satisfy the need of businesses and consumers. It helps businesses keep up with the quick urbanization and population growth, setup resource utilization and donate to city's economic development and stability.

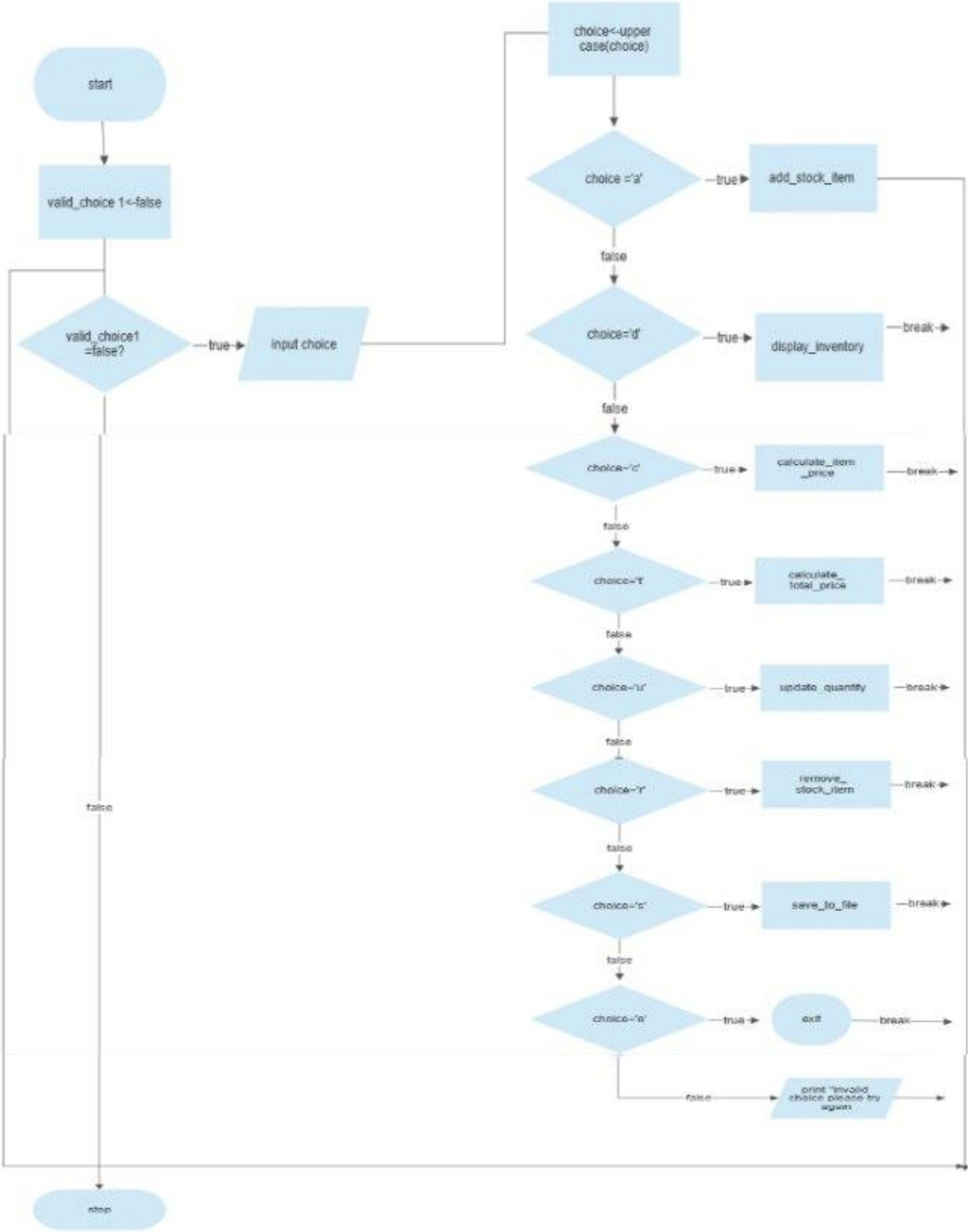
3.Pseudocode

```
Stock items = empty list

Function add_stock_item(inventory):
    Name = input(“Enter the name of the item: “)
    Quantity = input(“Enter the quantity of the item: “)
    Price = input(“Enter the price of the item: “)
    New item = create_stock_item(name, quantity, price)
    Append new_item to inventory
    Print “Item added successfully!”

Function display_ inventory(inventory):
    Print “Current Inventory:”
    For item in inventory:
        Print “Name: “ + item.name + “, Quantity: “ + item.quantity + “, Price: “ + item.price

Function calculate_item_prices(inventory):
    Print “Item Prices:”
    For item in inventory:
        Item_price = item.quantity * item.price
    Print "Name: " + item.name + ", Item Price: " + item_price

Function calculate_total_price(inventory):
  Total_price = 0

  For item in inventory:
    Item_price = item.quantity * item.price
    Total_price = total_price + item_price
     Return total_price

Function update_quantity(inventory):
  Name = input("Enter the name of the item to update: ")
  New_quantity = input("Enter the new quantity: ")
  Found = false
  For item in inventory:
    If item.name is equal to name:
      Item.quantity = new_quantity
      Found = true
      Break
  If found is true:
    Print "Quantity updated successfully!"
  Else:
    Print "Item not found!"

Function remove_stock_item(inventory):
    Name = input("Enter the name of the item to remove: ")
    Found = false
    For item in inventory:
        If item.name is equal to name:
            Remove item from inventory
            Found = true
            Break
    If found is true:
        Print "Item removed successfully!"
    Else:
        Print "Item not found!"

Function save_to_file(inventory):
    File = open("inventory.txt", "w")
    For item in inventory:
        Write item.name to file
        Write item.quantity to file
       Write item.price to file
Close file
    Print "Inventory saved to file successfully!"

Function read_from_file(inventory):
    File = open("inventory.txt", "r")
    For line in file:
        Split line into name, quantity, price
        Create new_item with name, quantity, price
        Append new_item to inventory
    Close file
    Print “Inventory loaded from file successfully!”
Function main():
    Inventory = empty list

    Choice = -1
    While choice is not equal to 9:
        Print “Stock Management System”
        Print “----------------------”
        Print “1. Add Stock Item”
        Print “2. Display Inventory”
        Print “3. Calculate Item Prices”
        Print “4. Calculate Total Price”
        Print “5. Update Quantity”
        Print “6. Remove Stock Item”
        Print “7. Save Inventory to File”
        Print “8. Load Inventory from File”
        Print “9. Exit”
        Choice = input(“Enter your choice: “)
        Switch choice:
            Case 1:
                Add_stock_item(inventory)
                Break
```

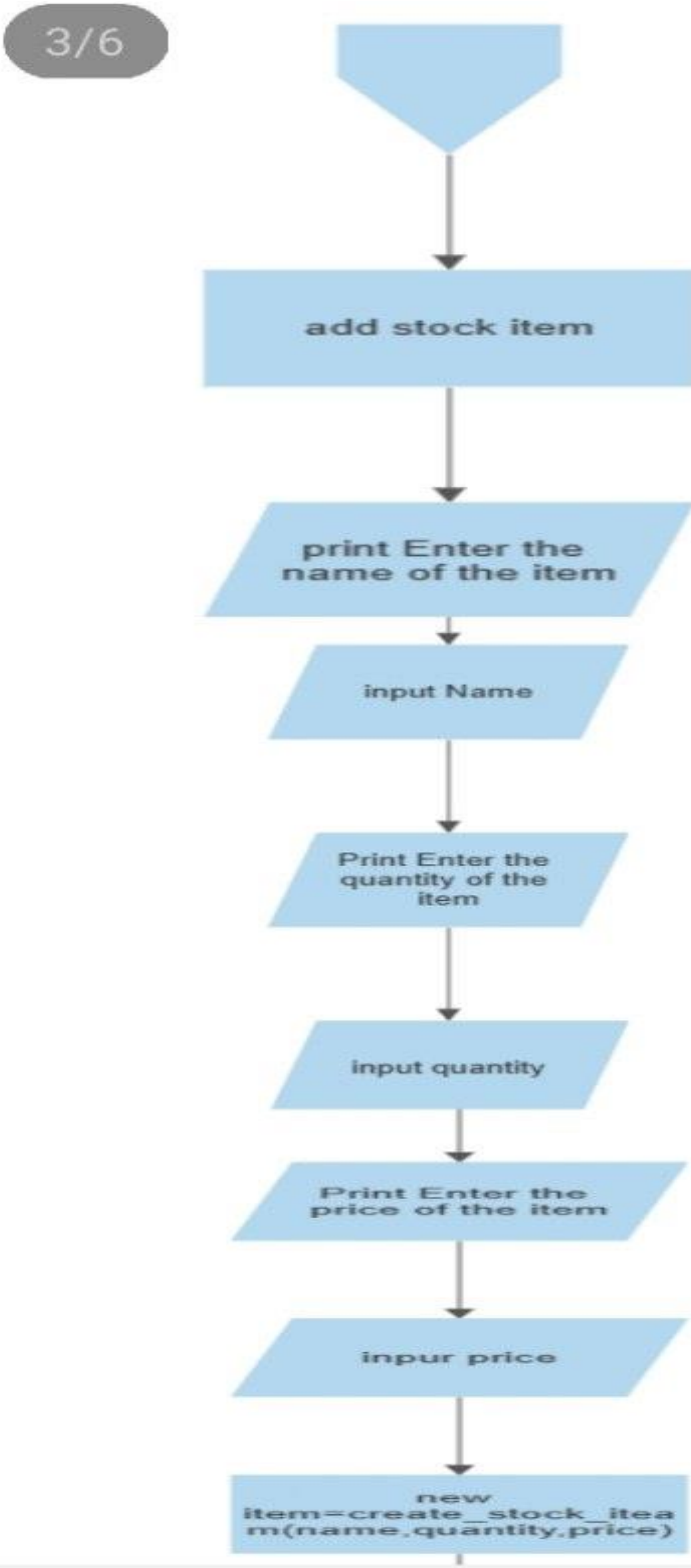
```
Case 2:
    Display_inventory(inventory)
    Break
Case 3:
    Calculate_item_prices(inventory)
    Break
Case 4:
    Total_price = calculate_total_price(inventory)
    Print “Total Price of all Items: “ + total_price
    Break
Case 5:
    Update_quantity(inventory)
    Break
Case 6:
    Remove_stock_item(inventory)
    Break
Case 7:
    Save_to_file(inventory)
    Break
Case 8:
    Read_from_file(inventory)
    Break
Case 9:
    Print “Exiting…”
    Break
```

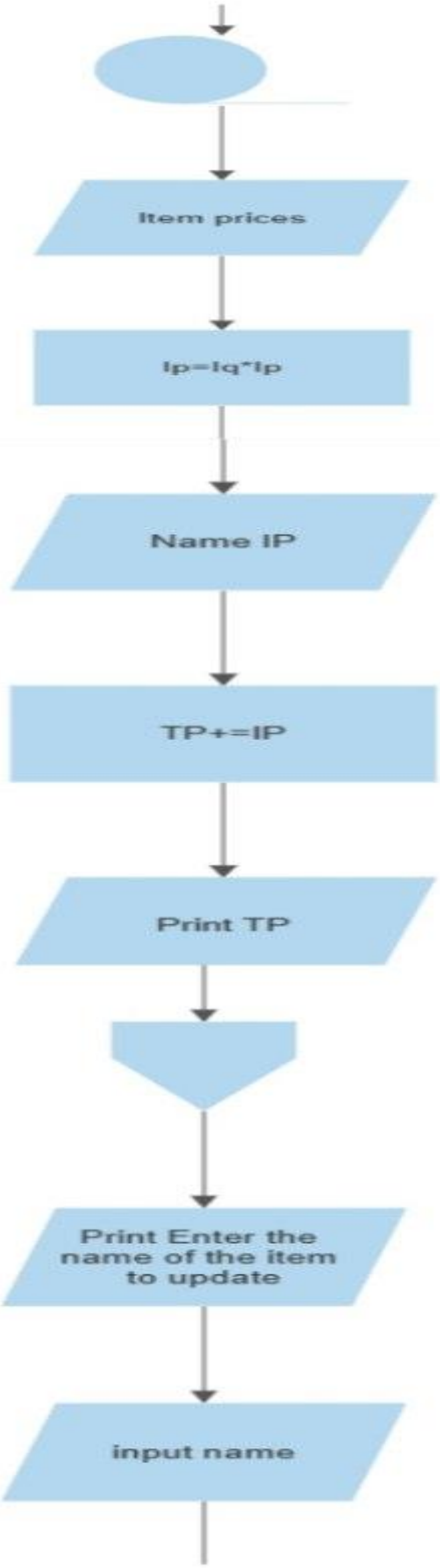
```
Default:
    Print "Invalid choice. Please try again."
```

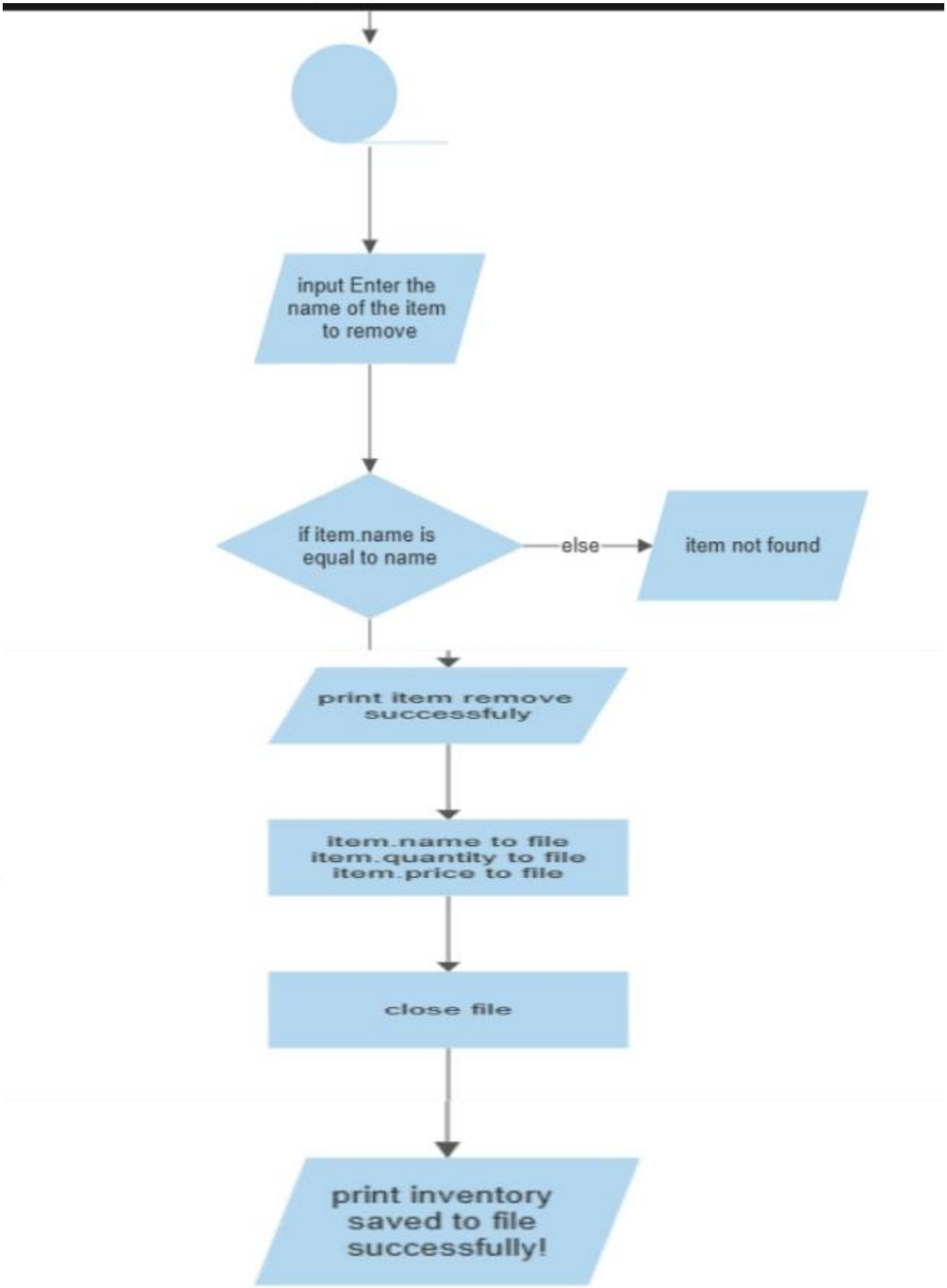
Flow chart


start
valid_choice 1<-false
valid_choice1 =false?
true
input choice
false
choice<-upper case(choice)
choice ='a'
true
add_stock_item
false
choice='d'
true
display_inventory
break
false
choice='c'
true
calculate_item _price
break
false
choice='t'
true
calculate_ total_price
break
false
choice='u'
true
update_quantity
break
false
choice='r'
true
remove_ stock_item
break
false
choice='s'
true
save_to_file
break
false
choice='e'
true
exit
break
false
print "invalid choice please try again
stop

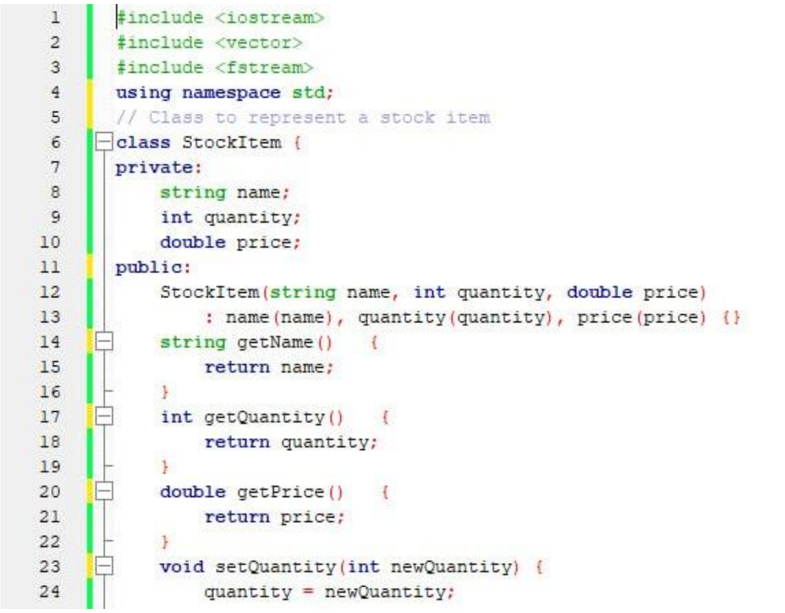
Stock management system


3/6
add stock item
print Enter the name of the item
input Name
Print Enter the quantity of the item
input quantity
Print Enter the price of the item
inpur price
new item=create_stock_iteam(name,quantity,price)

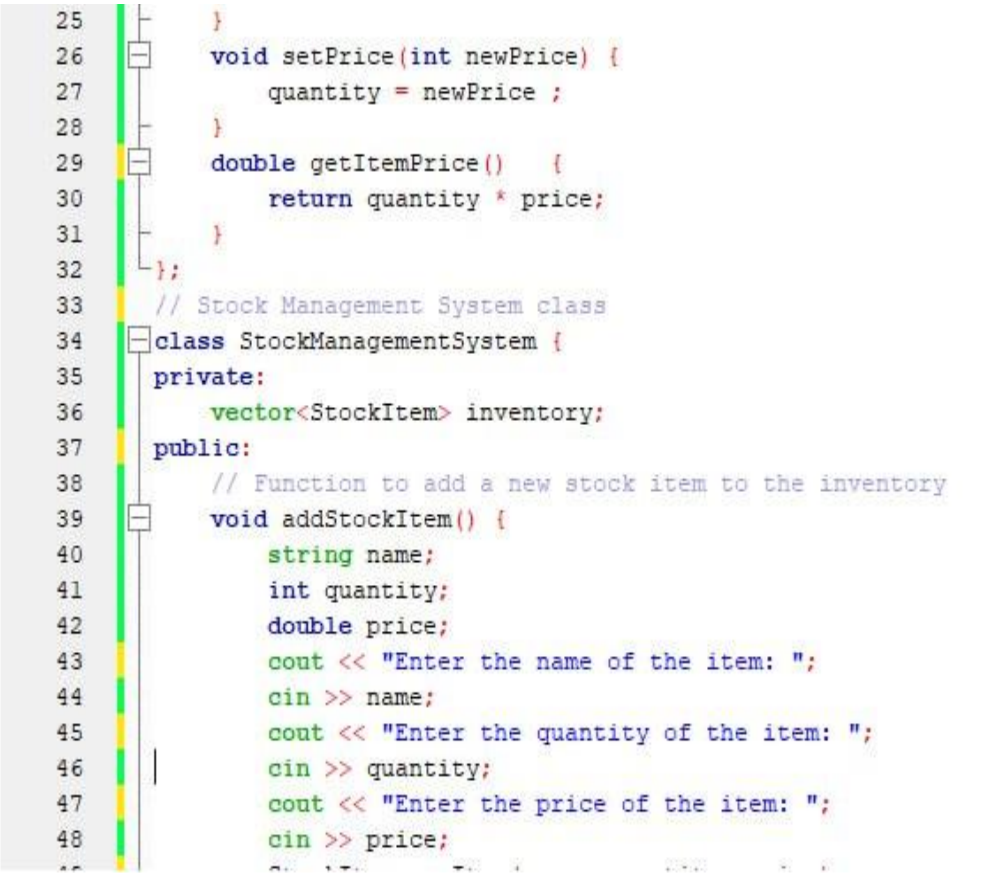

Item prices
Ip=Iq*Ip
Name IP
TP+=IP
Print TP
Print Enter the name of the item to update
input name


input Enter the name of the item to remove
if item.name is equal to name
else
item not found
print item remove successfuly
item.name to file
item.quantity to file
item.price to file
close file
print inventory saved to file successfully!

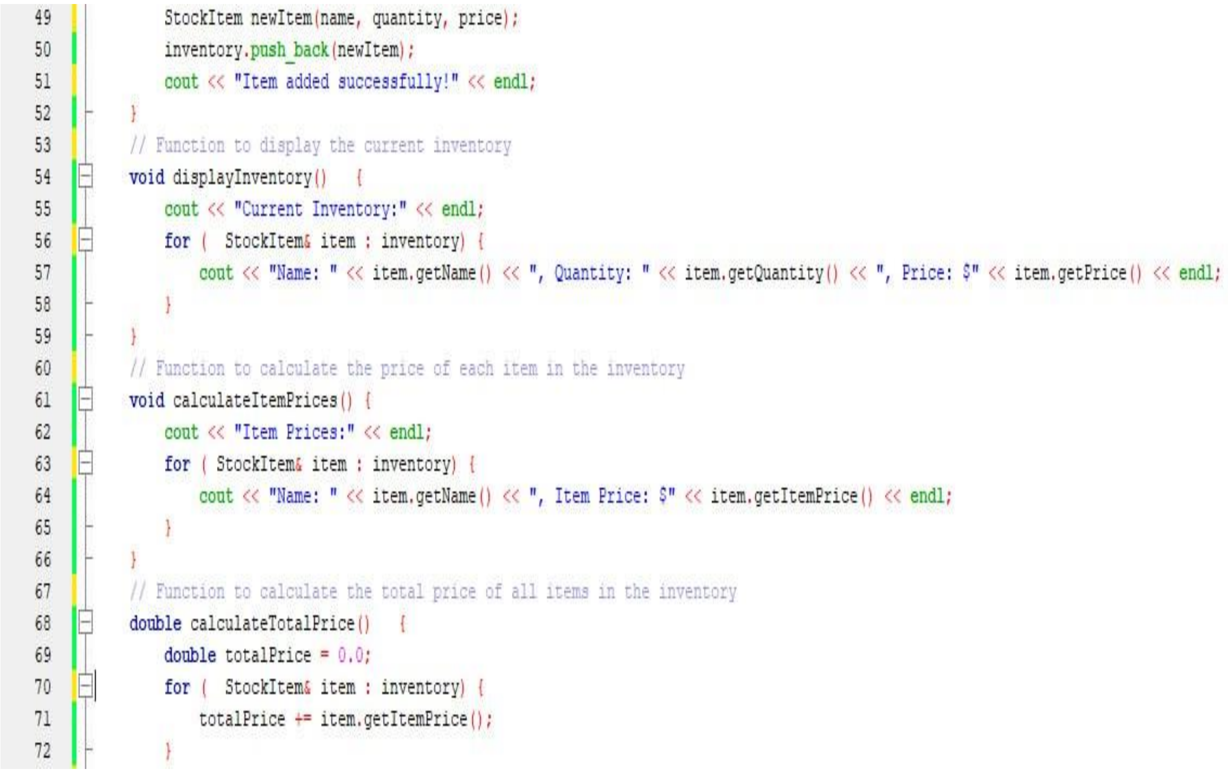
5.C++ console based application code & outputs



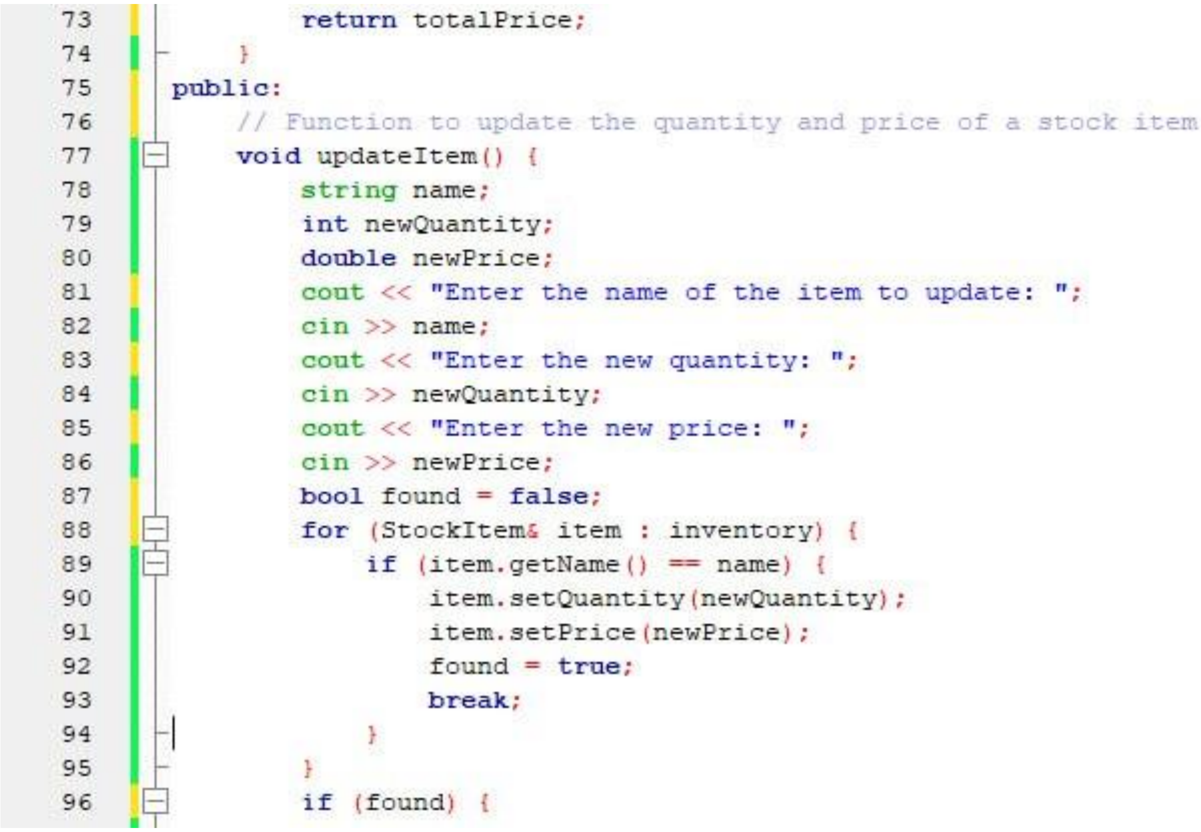
```
#include <iostream>
#include <vector>
#include <fstream>
using namespace std;
// Class to represent a stock item
class StockItem {
private:
    string name;
    int quantity;
    double price;
public:
    StockItem(string name, int quantity, double price)
        : name(name), quantity(quantity), price(price) {}
    string getName()    {
        return name;
    }
    int getQuantity()    {
        return quantity;
    }
    double getPrice()    {
        return price;
    }
    void setQuantity(int newQuantity) {
        quantity = newQuantity;
```



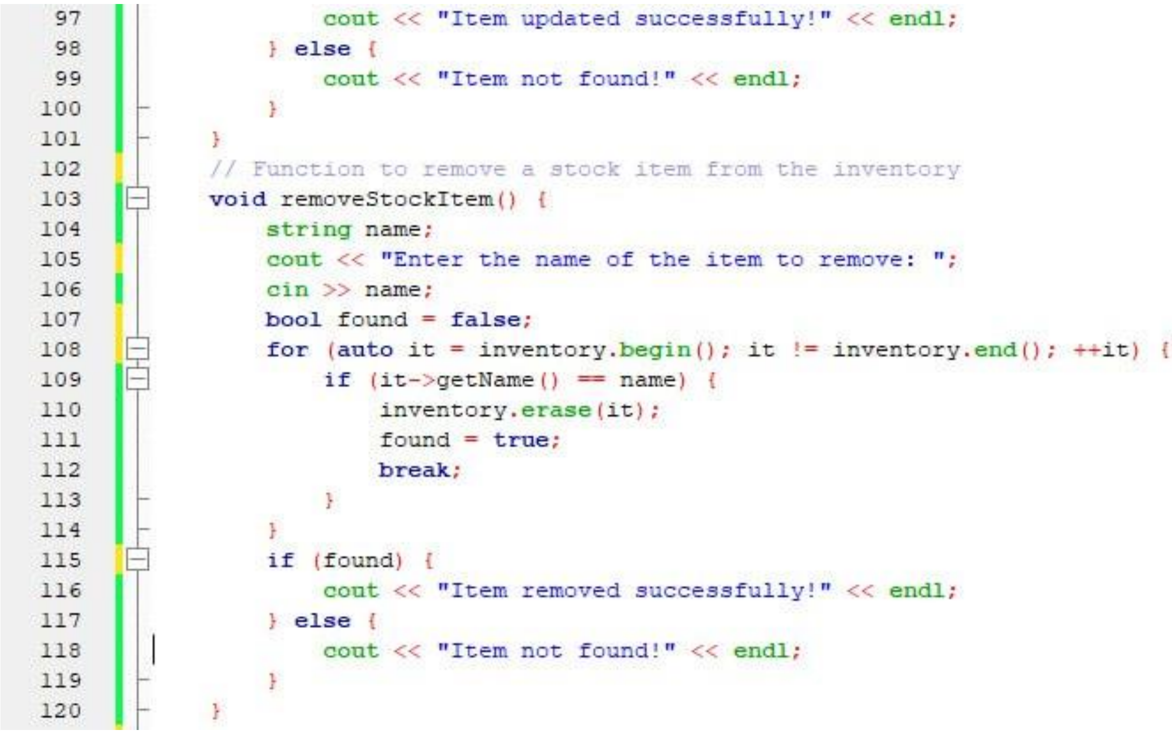
```
    }
    void setPrice(int newPrice) {
        quantity = newPrice ;
    }
    double getItemPrice()    {
        return quantity * price;
    }
};
// Stock Management System class
class StockManagementSystem {
private:
    vector<StockItem> inventory;
public:
    // Function to add a new stock item to the inventory
    void addStockItem() {
        string name;
        int quantity;
        double price;
        cout << "Enter the name of the item: ";
        cin >> name;
        cout << "Enter the quantity of the item: ";
        cin >> quantity;
        cout << "Enter the price of the item: ";
        cin >> price;
```



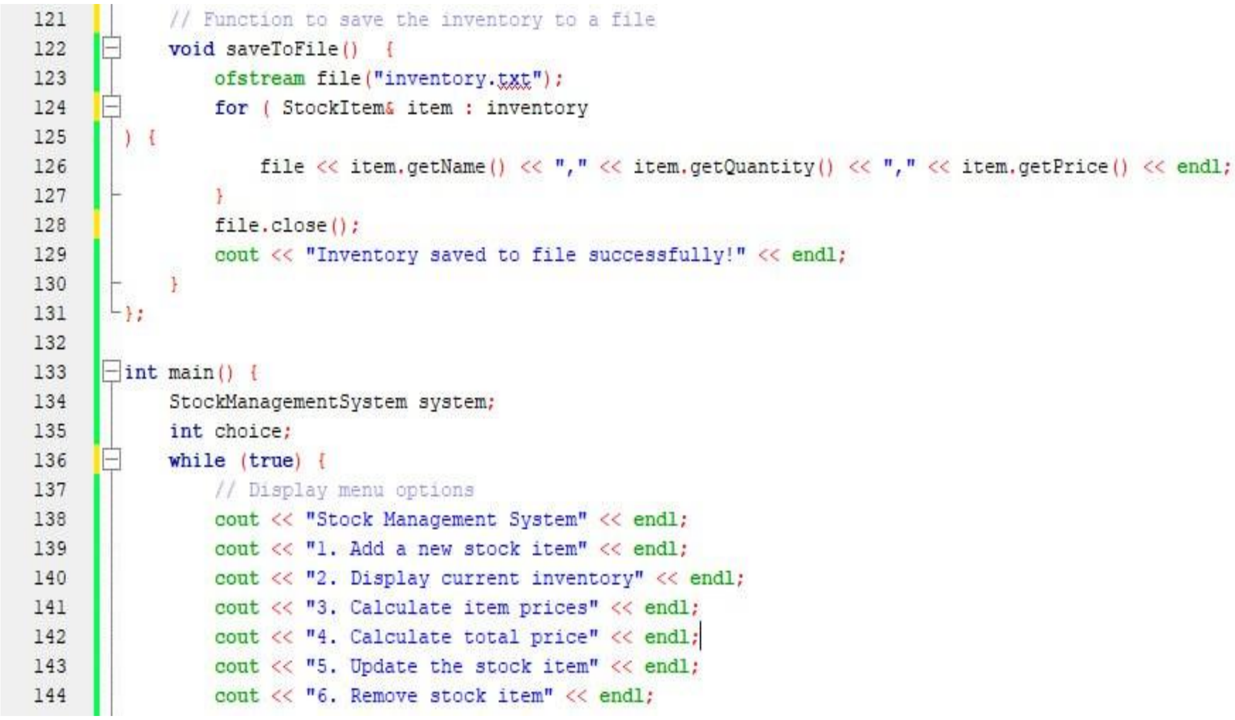
```cpp
        StockItem newItem(name, quantity, price);
        inventory.push_back(newItem);
        cout << "Item added successfully!" << endl;
    }
    // Function to display the current inventory
    void displayInventory()   {
        cout << "Current Inventory:" << endl;
        for (  StockItem& item : inventory) {
            cout << "Name: " << item.getName() << ", Quantity: " << item.getQuantity() << ", Price: $" << item.getPrice() << endl;
        }
    }
    // Function to calculate the price of each item in the inventory
    void calculateItemPrices() {
        cout << "Item Prices:" << endl;
        for ( StockItem& item : inventory) {
            cout << "Name: " << item.getName() << ", Item Price: $" << item.getItemPrice() << endl;
        }
    }
    // Function to calculate the total price of all items in the inventory
    double calculateTotalPrice()   {
        double totalPrice = 0.0;
        for (  StockItem& item : inventory) {
            totalPrice += item.getItemPrice();
        }
```



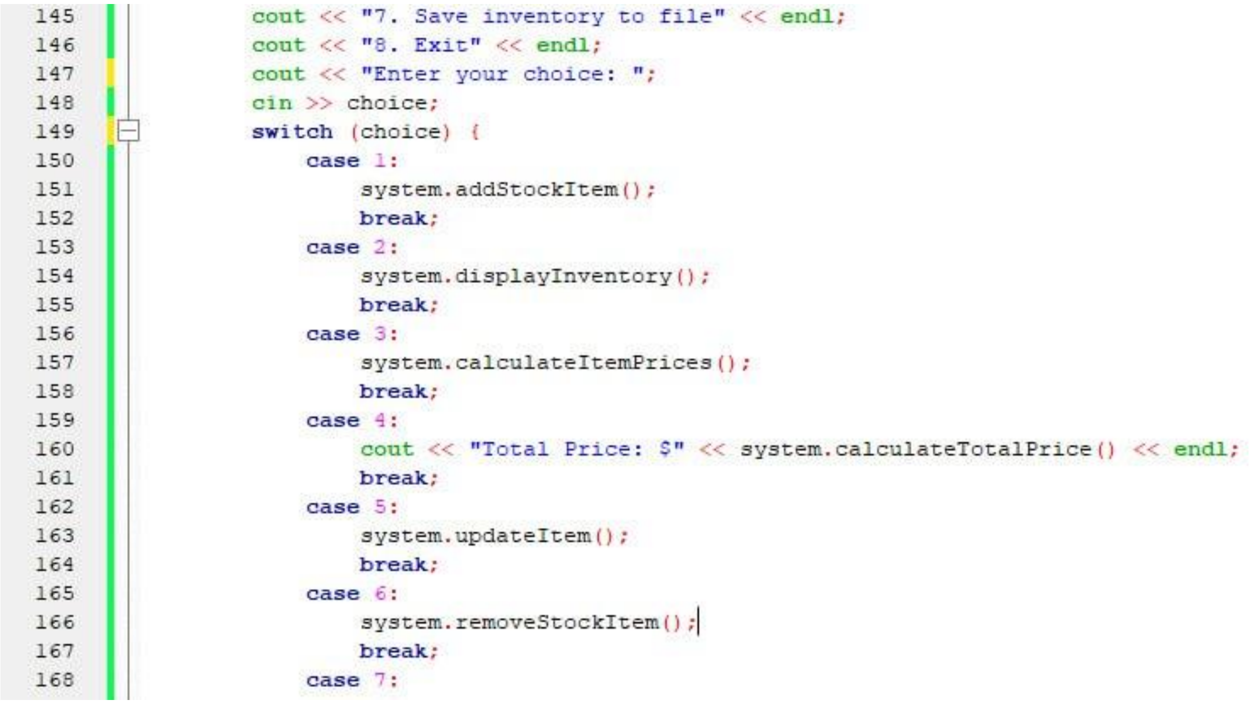
```cpp
        return totalPrice;
    }
public:
    // Function to update the quantity and price of a stock item
    void updateItem() {
        string name;
        int newQuantity;
        double newPrice;
        cout << "Enter the name of the item to update: ";
        cin >> name;
        cout << "Enter the new quantity: ";
        cin >> newQuantity;
        cout << "Enter the new price: ";
        cin >> newPrice;
        bool found = false;
        for (StockItem& item : inventory) {
            if (item.getName() == name) {
                item.setQuantity(newQuantity);
                item.setPrice(newPrice);
                found = true;
                break;
            }
        }
        if (found) {
```



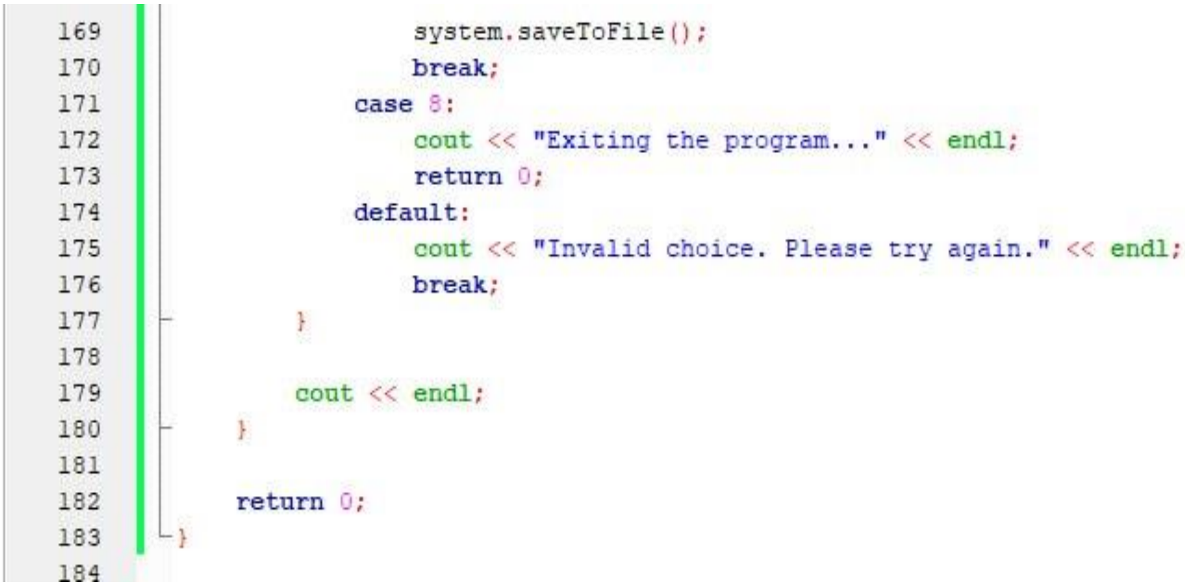
```
            cout << "Item updated successfully!" << endl;
        } else {
            cout << "Item not found!" << endl;
        }
    }
    // Function to remove a stock item from the inventory
    void removeStockItem() {
        string name;
        cout << "Enter the name of the item to remove: ";
        cin >> name;
        bool found = false;
        for (auto it = inventory.begin(); it != inventory.end(); ++it) {
            if (it->getName() == name) {
                inventory.erase(it);
                found = true;
                break;
            }
        }
        if (found) {
            cout << "Item removed successfully!" << endl;
        } else {
            cout << "Item not found!" << endl;
        }
    }
```



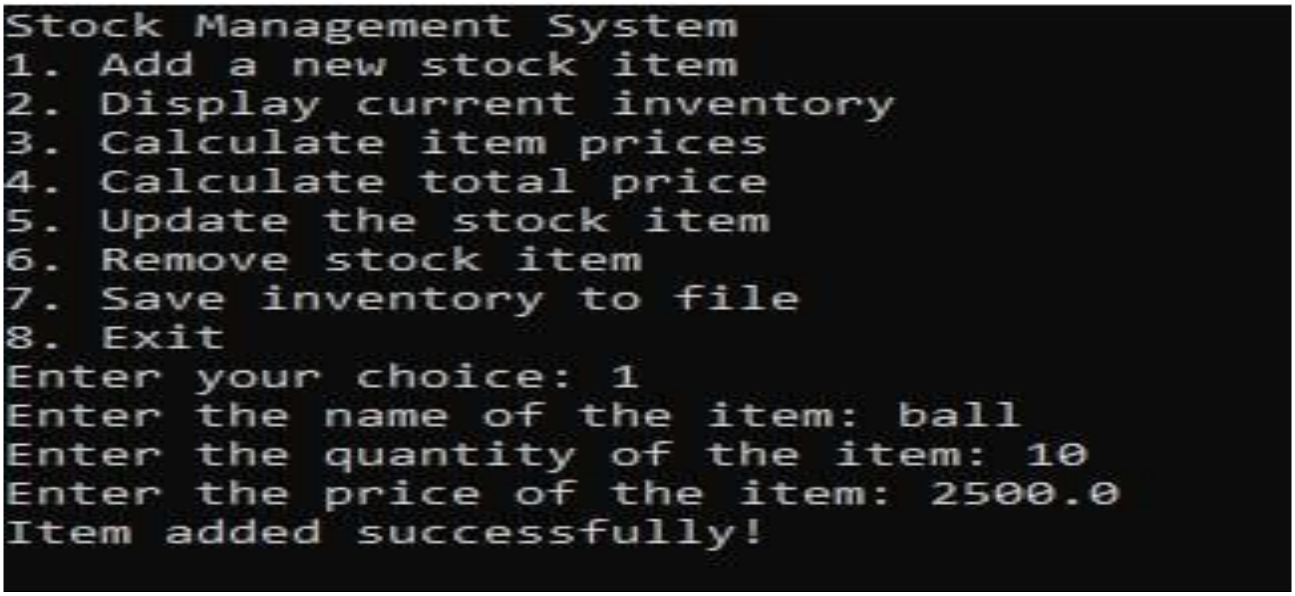
```
    // Function to save the inventory to a file
    void saveToFile()  {
        ofstream file("inventory.txt");
        for ( StockItem& item : inventory
) {
            file << item.getName() << "," << item.getQuantity() << "," << item.getPrice() << endl;
        }
        file.close();
        cout << "Inventory saved to file successfully!" << endl;
    }
};

int main() {
    StockManagementSystem system;
    int choice;
    while (true) {
        // Display menu options
        cout << "Stock Management System" << endl;
        cout << "1. Add a new stock item" << endl;
        cout << "2. Display current inventory" << endl;
        cout << "3. Calculate item prices" << endl;
        cout << "4. Calculate total price" << endl;
        cout << "5. Update the stock item" << endl;
        cout << "6. Remove stock item" << endl;
```



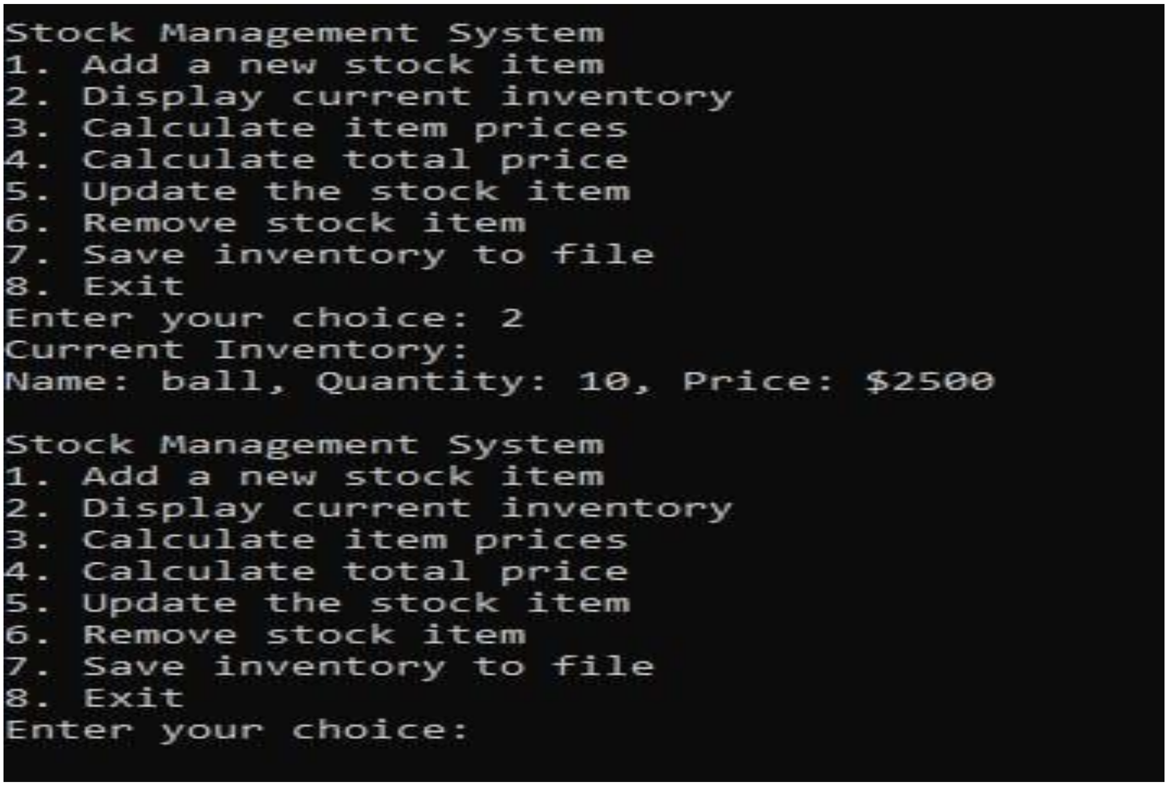
```
        cout << "7. Save inventory to file" << endl;
        cout << "8. Exit" << endl;
        cout << "Enter your choice: ";
        cin >> choice;
        switch (choice) {
            case 1:
                system.addStockItem();
                break;
            case 2:
                system.displayInventory();
                break;
            case 3:
                system.calculateItemPrices();
                break;
            case 4:
                cout << "Total Price: $" << system.calculateTotalPrice() << endl;
                break;
            case 5:
                system.updateItem();
                break;
            case 6:
                system.removeStockItem();
                break;
            case 7:
```



```
                system.saveToFile();
                break;
            case 8:
                cout << "Exiting the program..." << endl;
                return 0;
            default:
                cout << "Invalid choice. Please try again." << endl;
                break;
        }

        cout << endl;
    }

    return 0;
}
```



```
Stock Management System
1. Add a new stock item
2. Display current inventory
3. Calculate item prices
4. Calculate total price
5. Update the stock item
6. Remove stock item
7. Save inventory to file
8. Exit
Enter your choice: 1
Enter the name of the item: ball
Enter the quantity of the item: 10
Enter the price of the item: 2500.0
Item added successfully!
```



```
Stock Management System
1. Add a new stock item
2. Display current inventory
3. Calculate item prices
4. Calculate total price
5. Update the stock item
6. Remove stock item
7. Save inventory to file
8. Exit
Enter your choice: 2
Current Inventory:
Name: ball, Quantity: 10, Price: $2500

Stock Management System
1. Add a new stock item
2. Display current inventory
3. Calculate item prices
4. Calculate total price
5. Update the stock item
6. Remove stock item
7. Save inventory to file
8. Exit
Enter your choice:
```

6.Conclusion

The development of stock management system for Addis Ababa city using C++ and systems based on object oriented concepts is an important mechanism to increase productivity and effectiveness in the city's stock management system. The project's objective is to provide convenient and practical solution for businesses and organizations in Addis Ababa to efficiently manage their stocks, enhance security of inventory control.

Moreover, this stock management system will provide functionalities such as stock tracking, order management, supplier management, and report generation. These features will enable businesses and organizations to track their stock levels, easily place orders, manage their suppliers, and generate comprehensive reports for better decision-making.

Additionally this system will also focus on data security by implementing appropriate user access control levels and data backup mechanisms. This will protect sensitive company and customer information, ensuring the confidentiality and integrity of data.

Overall, the development of a stock management system for Addis Ababa city using C++ console-based application or systems using object-oriented concepts offers immense value to businesses and organizations in improving their stock management processes.

7.References

1. Weiss, M.A. (2013). Data Structures and Algorithm Analysis in C++. Boston, MA: Pearson Education.
2. Lippman, S.B., Lajoie, J., & Moo, B. (2012). C++ Primer (5th Ed.). Boston, MA: Addison-Wesley.
3. Stroustrup, B. (2013). The C++ Programming Language (4th Ed.). Boston, MA: Addison-Wesley.
4. Pressman, R.S. (2014). Software Engineering: A Practitioner's Approach (8th Ed.). New York, NY: McGraw-Hill.
5. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Reading, MA: Addison-Wesley.
6. Sommerville, I. (2015). Software Engineering (10th Ed.). Boston, MA: Pearson Education.
7. Rumbaugh, J., Jacobson, I., & Booch, G. (1999). The Unified Modeling Language Reference Manual (2nd Ed.). Reading, MA: Addison-Wesley.
8. Meyer, B. (1997). Object-Oriented Software Construction (2nd Ed.). Upper Saddle River, NJ: Prentice Hall.
9. Shelly, G.B., & Cashman, T.J. (2017). C++ Programming: From Problem Analysis to Program Design (8th Ed.). Boston, MA: Cengage Learning.
10. Deitel, P. J. (2017). How to Program C++ (2017 edition). Boston, MA: Pearson.