



**Green University of Bangladesh**  
**Department of Computer Science and Engineering (CSE)**  
**Faculty of Sciences and Engineering**  
**Semester: (Spring, Year:2021), B.Sc. in CSE (Day/Eve)**

**Course Title: CSE Lab**  
**Course Code: CSE 106**                      **Section: DD**

**Lab Project Name: STOCK MANAGEMENT SYSTEM**

**Student Details**

Name	ID
SAMIYA AKTER	213902044

**Submission Date: 9/10/2022**

**Course Teacher's Name: Md. Sultanul Islam Ovi**

**[For Teachers use only: Don't Write Anything inside this box]**

<b><u>Lab Project Status</u></b>	
<b>Marks: .....</b>	<b>Signature: .....</b>
<b>Comments: .....</b>	<b>Date: .....</b>

# Table of Contents

## **Chapter 1 Introduction**

- 1.1 Introduction
- 1.2 Design Goals/Objective

## **Chapter 2 Design/Development/Implementation of the Project**

- 2.1 Section (Choose the name of this section as appropriate with your project)
- 2.2 Section (Choose the name of this section as appropriate with your project)
  - 2.2.1 Subsection

## **Chapter 3 Performance Evaluation**

- 3.1 Simulation Environment/ Simulation Procedure
- 3.2 Results and Discussions

## **Chapter 4 Conclusion**

- 4.1 Introduction
- 4.1 Practical Implications
- 4.2 Scope of Future Work

## **References**

# Chapter 1

## Introduction

### 1.1 Introduction

The Stock Management System refers to the system and processes to manage the stock of an organization with the involvement of a Technology system. This system can be used to store the details of the stock, stock maintenance, update the stock based on the sales details, and generate sales and stock reports daily or weekly based. This project categorizes individual aspects for the sales and stock management system. In this system we are solving different problems affecting direct sales management and purchase management. Stock Management System is important to ensure quality control in businesses that handle transactions revolving around consumer goods. Without proper stock control, a large retail store may run out of stock on an important item. A good stock management system will alert the wholesaler when it is time to record. Stock Management System is also an important means of automatically tracking large shipments. An automated Stock Management System helps to minimize the errors while recording the stock.

### 1.2 Design Goals/Objective

- ❖ To know the fundamentals of the .Net Technology with the .
- ❖ To develop an application that deals with the day to day requirement of any production organization
- ❖ To develop the easy management of the inventory
- ❖ To handle the inventory details like sales details, purchase details and balance stock details.
- ❖ To provide competitive advantage to the organization.
- ❖ To provide detailed information about the stock balance. To make the stock manageable and simplify the use of inventory in the organization.



# Chapter 2

## Design/Development/Implementation of the Project

### 2.1 Design

### 2.2 PSEUDO CODE

#### HEADER FILES & VARIABLES

```
#include<stdlib.h>
#include<string.h>
#include<stdio.h>

struct stock
{
    int stockno;
    char name[100];
    char quantity[100];
    float price;
    struct stock *next;
} * head;
```

#### BINARY SEARCH FUNCTION

```
//binary search
struct stock* middle(struct stock* start,struct stock* last)
{
    if (start == NULL)
        return NULL;

    struct stock* slow = start;
    struct stock* fast = start -> next;
```

```

while (fast != last)
{
    fast = fast -> next;
    if (fast != last)
    {
        slow = slow -> next;
        fast = fast -> next;
    }
}

return slow;
}
struct stock* binarySearch(struct stock *head, int stockno)
{
    struct stock* start = head;
    struct stock* last = NULL;

    do
    {
        struct stock* mid = middle(start, last);

        if (mid == NULL)
            return NULL;

        if (mid -> stockno == stockno)
            return mid;

        else if (mid -> stockno < stockno)
            start = mid -> next;

        else
            last = mid;
    } while (last == NULL ||
            last != start);

    return NULL;
}

```

## INSERT

```

void insert(int stockno, char* name, char* quantity, float price)
{
    struct stock * stock = (struct stock *) malloc(sizeof(struct stock));
    stock->stockno = stockno;
    strcpy(stock->name, name);
    strcpy(stock->quantity, quantity);
    stock->price = price;
    stock->next = NULL;

    if(head==NULL){
        // if head is NULL
        // set stock as the new head
        head = stock;
    }
    else{
        // if list is not empty
        // insert stock in beginning of head
        stock->next = head;
        head = stock;
    }
}

```

## SEARCH

```

void search(int stockno)
{
    struct stock * temp = head;

    if(binarySearch(head,stockno)==NULL)
    {
        printf("Data not found");
    }
    else
    {
        printf("Stock Number: %d\n", temp->stockno);
        printf("Name: %s\n", temp->name);
        printf("quantity: %s\n", temp->quantity);
        printf("price: %0.4f\n", temp->price);
        return;
    }
}

```

```
}  
  
}
```

## UPDATE

```
void update(int stockno)
{
    struct stock * temp = head;
    while(temp!=NULL){
    if(temp->stockno==stockno){
        printf("Record with stock number %d Found !!!\n", stockno);
        printf("Enter new name: ");
        scanf("%s", temp->name);
        printf("Enter new quantity number: ");
        scanf("%s", temp->quantity);
        printf("Enter new price: ");
        scanf("%f",&temp->price);
        printf("Updation Successful!!!\n");
        return;
    }
    temp = temp->next;

    }
    printf("stock with stock number %d is not found !!!\n", stockno);
}
```

## DELETE

```
void Delete(int stockno)
{
    struct stock * temp1 = head;
    struct stock * temp2 = head;
    while(temp1!=NULL){

        if(temp1->stockno==stockno){

            printf("Record with stock number %d Found !!!\n", stockno);

            if(temp1==temp2){
                // this condition will run if
```



```

        // the record that we need to delete is the first node
        // of the linked list
        head = head->next;
        free(temp1);
    }
    else{
        // temp1 is the node we need to delete
        // temp2 is the node previous to temp1
        temp2->next = temp1->next;
        free(temp1);
    }

    printf("Record Successfully Deleted !!!\n");
    return;

}

temp2 = temp1;
temp1 = temp1->next;

}
printf("stock with stock number %d is not found !!!\n", stockno);
}

```

## DISPLAY

```

void display()
{
    struct stock * temp = head;
    while(temp!=NULL){

        printf("Stock Number: %d\n", temp->stockno);
        printf("Name: %s\n", temp->name);
        printf("quantity: %s\n", temp->quantity);
        printf("price: %0.4f\n\n", temp->price);
        temp = temp->next;
    }
}

```

## SORTING

```

void sort()
{
    struct stock *curNode,*nextNode;
    curNode = head;
    while(curNode!=0)
    {
        nextNode = curNode->next;

        while(nextNode!=0)
        {
            if(curNode->stockno>nextNode->stockno)
            {
                swap1(&curNode->stockno, &nextNode->stockno);
                swap2(curNode->name, nextNode->name);
                swap2(curNode->quantity, nextNode->quantity);
                swap3(&curNode->price, &nextNode->price);
            }
            nextNode = nextNode->next;
        }
        curNode = curNode->next;
    }
    printf("\nRecord is Now Sorted\n");
}

```

## SWAP

```

void swap1( int *a, int *b)
{
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}

void swap3(float *x, float *y)
{
    float temp;
    temp = *x;
    *x = *y;
}

```

```

    *y = temp;
}

void swap2(char *str1, char *str2)
{
    char *temp = (char *)malloc((strlen(str1) + 1) * sizeof(char));
    strcpy(temp, str1);
    strcpy(str1, str2);
    strcpy(str2, temp);
    free(temp);
}

```

## MAIN FUNCTION

```

int main()
{
    head = NULL;
    int choice;
    char name[100];
    char quantity[100];
    int stockno;
    float price;
    printf("1 to insert stock details\n2 to search for stock details\n3 to delete stock
details\n4 to update stock details\n5 to display all stock details");
    do
    {
        printf("\nEnter Choice: ");
        scanf("%d", &choice);
        switch (choice)
        {
            case 1:
                printf("Enter stock number: ");
                scanf("%d", &stockno);
                printf("Enter name: ");
                scanf("%s", name);
                printf("Enter quantity number: ");
                scanf("%s", quantity);
                printf("Enter price: ");
                scanf("%f", &price);
                insert(stockno, name, quantity, price);

```

```

        break;

    case 2:
        printf("Enter stock number to search: ");
        scanf("%d", &stockno);
        search(stockno);
        break;

    case 3:
        printf("Enter stock number to delete: ");
        scanf("%d", &stockno);
        Delete(stockno);
        break;

    case 4:
        printf("Enter stock number to update: ");
        scanf("%d", &stockno);
        update(stockno);
        break;

    case 5:
        display();
        break;
    }

} while (choice != 0);
}

```

## Chapter 3

### Performance Evaluation

#### 3.1 OUTPUT

##### ENTER CHOICE

"E:\3rd semister cse lab\stock management system\STOCK MANAGEMENT SYSTEM.exe"

```
1 to insert stock details
2 to search for stock details
3 to delete stock details
4 to update stock details
5 to display all stock details
Enter Choice: _
```

## INSERT

```
Enter Choice: 1
Enter stock number: 01
Enter name: IPHONE
Enter quantity number: 6
Enter price: 100000
```

## SEARCH

```
Enter Choice: 2
Enter stock number to search: 01
Stock Number: 1
Name: IPHONE
quantity: 6
price: 100000.0000
```

## DISPLAY

```
Enter Choice: 5
Stock Number: 1
Name: IPHONE
quantity: 6
price: 100000.0000
```

### **UPDATE**

```
Enter Choice: 4
Enter stock number to update: 01
Record with stock number 1 Found !!!
Enter new name: LAPTOP
Enter new quantity number: 2
Enter new price: 400000
Updation Successful!!!
```

### **DELETE**

```
Enter Choice: 3
Enter stock number to delete: 01
Record with stock number 1 Found !!!
Record Successfully Deleted !!!
```

## **3.2 Results and Discussions**

### **Analysis and Outcome**

By using a stock management system we can easily insert a product detail. We can know the quantity and price of the product. WE can Know the data of our stock such as how many products are there in our stock, what is the price, what is the name of that product etc.

## **Chapter 4**

## **Conclusion**

**It is the best project for beginners. In this project, you can check the stock of a particular store and can search the items with its quantity. It has no graphics but can be used for practical purposes.**

## **4.1 Scope of Future Work**

**In the future it can help- us develop our business or store. We can take early preparation to maintain our stock full with the quantity of product we need.**

## **References**

**[1] Used google to get the basic structure of my code and the algorithm.**

**[2] Got the idea for slideshare.com**