

TABLE OF CONTENTS

ABSTRACT	6
1. INTRODUCTION	7
1.1 PROJECT AIMS AND OBJECTIVES	7
1.2 BACKGROUND OF PROJECT	8
1.3 OPERATION ENVIRONMENT	9
2. SOFTWARE AND HARDWARE REQUIREMENTS	10
3. SYSTEM IMPLEMENTATION & TESTING	11
4. CODE	15
5. CONCLUSION	24
6. REFERENCES	25

ABSTRACT

Shopping Club management system is a project which aims in developing a computerized system to maintain all the daily work of Shopping Club.

AIM: To provide a comprehensive solution for the shopping club management system.

As we all know, people have been relying on the shopping very much. Simple Shopping Club Management System is based on the concept of managing the record of shopping club on regarding their items and customers. This is a better design of a system wherein people can easily retrieve info without wasting much time.

"Shopping Club Management System" has been designed to overcome the shopping problems.

CHAPTER 1

INTRODUCTION

This chapter gives an overview about the aim, objectives, background and operation environment of the system.

1.1 PROJECT AIMS AND OBJECTIVES

The project aims and objectives that will be achieved after completion of this project are discussed in this subchapter. The aims and objectives are as follows:

- To create a Shopping Club management system that is user-friendly.
- The main objectives of the Shopping Club Management System is to manage the details of products, shopping, and customers.
- The software is capable enough to allow the concerned person to store and retrieve any type of product record. The software allows interactive, self-describing Graphic User Interface environment where even standalone users can work very comfortably and easily.
- All the data pertaining to product information is kept at central database from where it can be easily retrieved, viewed or updated. But, such kind of technical details are hidden from the standalone user. He just needs to select the relevant option from the given menu-driven interface. However, the central repository of data can be easily accessed if required
- Data Redundancy is no more the problem now. The data modified from one particular data entry form will reflect at central database as well. The software ensures that all duplicate products are efficiently managed.
- The software provides the facility of storing new products and also displays the existing ones in order.

1.2 BACKGROUND OF PROJECT

This system is developed keeping in view the general need required by the Customer while doing the shopping. To keep the items and other data updated, the admin will have the authority to add and delete as well as modify the existing records within the shopping club. The admin of the market will have the authority to list details of all available items.

The admin can also perform various operations such as adding customer records, deleting records, modifying existing records etc. The Customer can choose trollies for shopping and will get a bill at the end of shopping.

To make all operations as easier as possible, a user-friendly approach has been taken into account by which users have to only give the relevant inputs during final confirmation to make their operations successful. The background processing system will take care of all processing tasks and maintain data integrity to reduce the redundancy of data. For shopping operation, customer will be able to get any particular product using their item id but the only condition is that product records must be available within the file system. If no such record is available, a proper error message will be displayed as per user input provided to the system.

1.3 OPERATION ENVIRONMENT

PROCESSOR	INTEL CORE PROCESSOR OR BETTER PERFORMANCE
OPERATING SYSTEM	WINDOWS , UBUNTU,MAC
MEMORY	1GB RAM OR MORE
HARD DISK SPACE	MINIMUM 3 GB FOR USAGE FOR FUTURE
SOFTWARE	DEV C++

2. SOFTWARE AND HARDWARE REQUIREMENTS

This section describes the software and hardware requirements of the system

SOFTWARE REQUIREMENTS

- Operating system- Windows 10 is used as the operating system as it is stable and supports more features and is more user friendly
- Software –DEV C++ is used to write code.
- Programming language- C++ is used as programming language.

HARDWARE REQUIREMENTS

- Intel core i5 2nd generation is used as a processor because it is fast than other processors and provide reliable and stable and we can run our pc for longtime. By using this processor we can keep on developing our project without any worries.
- Ram 1 GB is used as it will provide fast reading and writing capabilities and will in turn support in processing

❖ Data Structures Used

For Administrator Module - LINKED LIST, ARRAY

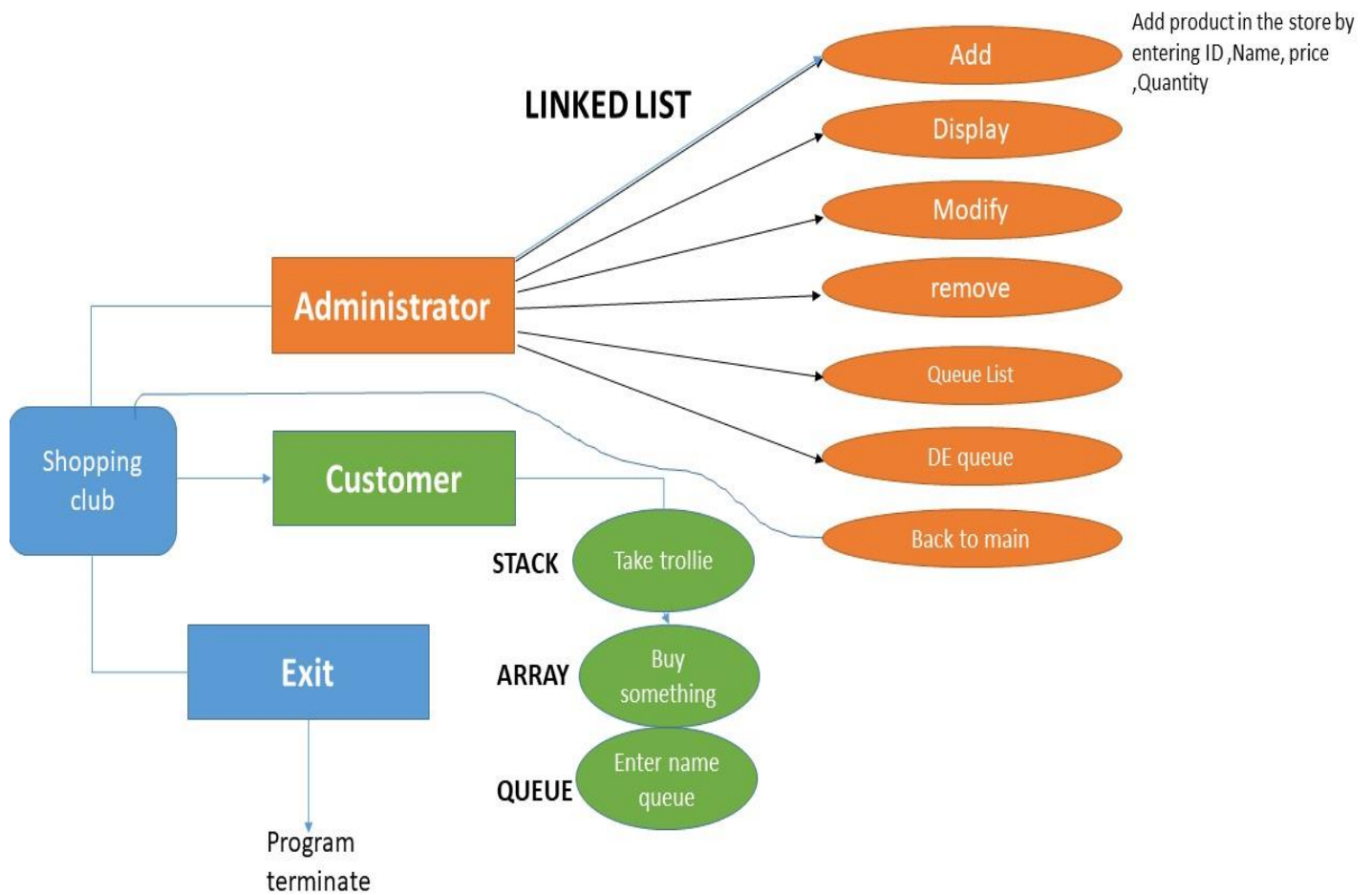
For Customer Module - ARRAY, STACK, QUEUE

1) Administrator Module uses Linked list to add, delete or modify products, display customers or remove customers.

2) Customer Module uses Array to buy something, Stack to take trollies, Queue to add name into queue

CHAPTER 3

SYSTEM IMPLEMENTATION & TESTING



❖ MAIN PAGE

```
C:\Users\Admin\Downloads\Shopping System C++ Source Code\Shopping System C++ Source Code\shopping.exe

-----
||      Shopping Club Management System      ||
-----

[Subject Teacher ->> DHAWALEKSHMI MAM & VIKASH SIR <<- ]

>>----Project Implemented By-----<<

CHIRAG SHARMA(LATG10024)
PRAKHAR JAIN(LATG10025)
ARYAN GARG(LATG10046)

Press any key to continue . . .
```

❖ MAIN MENU

```
C:\Users\Admin\Downloads\Shopping System C++ Source Code\Shopping System C++ Source Code\shopping.exe

]-----[Main Menu]-----[

1)Administrator of Market
2)Customer
3)Exit

Enter Your choice >>>
```


❖ ADMIN MODULE

```
C:\Users\Admin\Downloads\Shopping System C++ Source Code\Shopping System C++ Source Code\shopping.exe

=====
|      Administrator Portal      |
=====
Enter 1 for ADD a new product
Enter 2 to display all products
Enter 3 for MODIFY Existing product
Enter 4 for Delete a particular product item
Enter 5 for Customers List
Enter 6 for Dequeue customer
Enter 7 for Generate hash
Enter 0 for Main Menu

Enter Your choice >>>_
```

❖ CUSTOMER MODULE

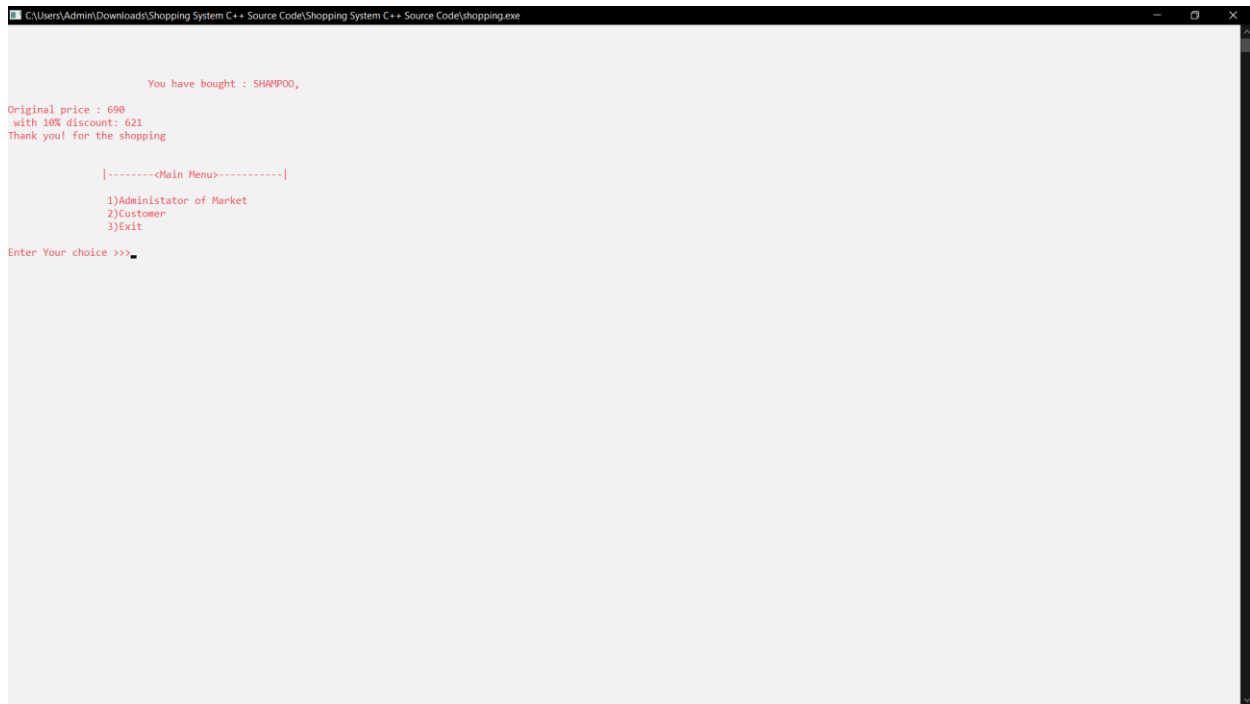
```
C:\Users\Admin\Downloads\Shopping System C++ Source Code\Shopping System C++ Source Code\shopping.exe

Existing products are:
=====
ID      Product Name      Price      Quantity
=====
1      SHAMPOO      230      5
2      OIL      100      3
=====

Total products in our store is : 2

How many itees you wanna to buy!
_
```

❖ BILLING PAGE



```
C:\Users\Admin\Downloads\Shopping System C++ Source Code\Shopping System C++ Source Code\shopping.exe

You have bought : SHAMPOO,

Original price : 690
with 10% discount: 621
Thank you! for the shopping

-----Main Menu-----
1)Administator of Market
2)Customer
3)Exit

Enter Your choice >>> _
```

- **Features of the project**

‘Shopping Club Management System’ has been designed to computerize the following Functions that are performed by the system under two modules:

1) Admin module (Administration of market)

- Add new products
- Display all products
- Modify existing products
- Delete any product
- List all the customers
- Remove any customer from list

2) Customer module (Customer)

- To get trollies for shopping
- To Order items

CODE

```

#include<iostream>

#include<string>
#include <sstream>
#include <bits/stdc++.h>
#include<windows.h>
#include"queue1.h"
#include"animation.h"
#include"stackme.h"

using namespace std;

int search(int);
int display();
string check(int);    // for checking quantity

////////////////////////////////////
struct node
{
    int ID;
    string proName;
    double prePrice;
    int quantity;
    struct node* next;
};

struct node *head=NULL;

////////////////////////////////////
void beg()
{
    system("cls");
    int id,quant;    // quant for quantity
    string name;
    double pre;    // pre for price
    struct node *t=new node;
    struct node *p=head;

    cout<<"\t\t\tEnter product ID:-";
    cin>>id;
    t->ID=id;
    cout<<"\t\t\tEnter product Name:-";
    cin>>name;
    t->proName=name;
    cout<<"\t\t\tEnter product price:-";
    cin>>pre;

```

```

t->prePrice=pre;
cout<<"\t\tEnter product quantity:-";
cin>>quant;
t->quantity=quant;
if(head==NULL)
{
t->next=head;
head=t;
}
else
{
while(p->next!=NULL)
{
p=p->next;
}
p->next=t;
t->next=NULL;
}
system("cls");
cout<<"\n\n\t\t\tThis product is Inserted!\n\n\n";
}

```

```

////////////////////////////////////
int search(int id)          // for search item in list
{
int count=1;
struct node *p=head;
while(p!=NULL)
{
if(p->ID==id)
break;
else
count++;
p=p->next;
}
return count;
}

```

```

////////////////////////////////////
int hash(int x, int mod)
{
return x % mod;
}

```

```

////////////////////////////////////
void delPro()
{
system("cls");
display();
}

```

```

        int id;
        struct node *cur=head;
        struct node *pre=head;
        cout<<"\n\nEnter ID to delete that product:\n\n";
        cin>>id;
        if (head == NULL)
    {
        system("cls");
        cout<<"List is empty"<<endl;
    }

    int pos=0;
    int count=display();          // for load no of nodes
    pos=search(id);              // for check weather desire node is exist or not
    if(pos<=count){

        while(cur->ID!=id){      // for delete middle area products
            pre=cur;
            cur=cur->next;
        }

        pre->next=cur->next;
        system("cls");
        cout<<"\n<<item is deleted>>\n";
    }else{
        cout<<"\n<<<Not found>>\n\n";
    }
}

////////////////////////////////////
void modify()
{
    int id;
    double pre;    // pre for price
    string pName;
    int nid; int nq; // pName for new name
    if (head == NULL)
    {
        system("cls");
        cout<<"List is empty"<<endl;
    }
    else
    {
        display();
        cout<<"\n\nEnter ID to modify product Name and its price:\n";
        cin>>id;
        struct node *cur=head;
        int pos=0;
        int count=display();      // for load no of nodes
        pos=search(id);
        // for check weather desire node is exist or not
        if(pos<=count)
    {

```

```

        while(cur->ID!=id)
        {
            cur=cur->next;
        }
        cout<<"\nOld ID : "<<cur->ID<<endl;
        cout<<"\nOld Name : "<<cur->proName<<endl;
        cout<<"\nOld Price : "<<cur->prePrice<<endl;
        cout<<"\nOld Quantity : "<<cur->quantity<<endl;

        cout<<endl<<endl;
        cout<<"Enter new ID:";
        cin>>nid;
        cur->ID=nid;
        cout<<"Enter new Name:";
        cin>>pName;
        cur->proName=pName;
        cout<<"Enter new Price:";
        cin>>pre;
        cur->prePrice=pre;
        cout<<"Enter new Quantity:";
        cin>>nq;
        cur->quantity=nq;
    }

    else
    {
        cout<<id<<" is <<<Not found>>\n\n";
    }
}

/////////////////////////////////////////////////////////////////

int display()
{
    system("cls");
    int c=0;          // c for count products
    struct node *p=head;
    cout<<"Existing products are:\n";
    cout<<"ID\tProduct Name\tPrice\tQuantity\n";
    cout<<"=====||\n";
    while(p!=NULL)
    {
        cout<<p->ID<<"\t"<<p->proName<<"\t"<<p->prePrice<<"\t"<<check(p-
>quantity)<<"\n"; // call check func and pass quantity
        p=p->next;
        c=c+1;
    }
    cout<<"\nTotal products in our store is : "<<c<<"\n\n";
    return c;
}

```

```

////////////////////////////////////
    string check(int quant)

    {
        //    check function
        int a = quant;
        stringstream ss;
        ss << a;
        string quantity = ss.str();

        if(quant<=0)
            return "out of stock!";
        else
            return quantity;
    }

////////////////////////////////////
    void buy()
    {
        system("cls");
        display();
        string products[20];
        // for display sold items
        int pay=0,no,c=0,price,id,i=1;

        if(head==NULL)
        {
            cout<<"\n<<<<There is no items to buy>>>>\n\n";
        }
        else
        {
            cout<<"How many items you wanna to buy!\n";
            cin>>no;
            int count=display();    // for store no of nodes in c
            while (i<=no)
            {
                struct node *cur=head;
                int quant,cho; a:    // quant for quantity and cho for choice
                cout<<"Enter id of item that you want to buy: ";
                int id,pos=0;
                cin>>id;
                if(id==-1){system("cls"); return; }
                pos=search(id);
                if(pos<=count)
                {
                    //    item is available in store
                    while(cur->ID!=id)
                    {
                        cur=cur->next;
                    }
                    cout<<"How many quantities you want:";
                    cin>>quant;

```

```

////////////////////////////////////
void administrator()
{
    system("cls");
    int ch;

    do {

```



```
// choice for below message
```

```
cout<<"\t\t===== "<<endl;  
cout<<"\t\t    Administator Portal    |"<<endl;  
cout<<"\t\t===== "<<endl;
```

```
cout<<"\t\t Enter 1 for ADD a new product "<<endl;
cout<<"\t\t Enter 2 to display all products "<<endl;
cout<<"\t\t Enter 3 for MODIFY Existing product"<<endl;
cout<<"\t\t Enter 4 for Delete a particular product item"<<endl;
cout<<"\t\t Enter 5 for Customers List "<<endl;
cout<<"\t\t Enter 6 for Dequeue customer"<<endl;
cout<<"\t\t Enter 7 for Generate hash"<<endl;
cout<<"\t\t Enter 0 for Main Menu"<<endl;
```

```
cout<<"\nEnter Your choice >>>"; cin>>ch;
switch(ch){
case 1:
beg();
break;
```

```
case 2:
    system("cls");
    display();
    break;
```

```
case 3:
    modify();
    system("cls");
    break;
```

```

case 4:
    delPro();
    cout<<"\n-----Product is Deleted-----\n";
    break;

```

```
case 5:
    system("cls");
    cout<<"|=====CUSTOMERS NAMES LIST=====|"<<endl;
    displayQueue();
    break;
```

```
case 6:
    system("cls");
    cout<<"=====CUSTOMERS NAMES LIST=====|"<<endl;
    dequeue();
    displayQueue();
    break;
```

```
case 7:
    int x,n;
    cout << "Enter element to generate hash = ";
    cin >> x; cout<<"Of total list number : "; cin>>n;
    cout << "Hash of " << x << " is " << hash(x,n);
```

```

        break;

default: system("cls");

    }
}
while(ch!=0)    ;
}

//      Main function
int main()
{
    for(int i=0;i<=51;i++)
    {
        push(i);
    }

    system("color F2");    // for console color
    gotoxy(17,5);
    cout<<"-----"<<endl;
    gotoxy(17,7);
    cout<<"||      Shopping Club Management System      ||"<<endl;
    gotoxy(17,9);
    cout<<"-----"<<endl;
    gotoxy(17,11);
    cout<<"|Subject Teacher ->> DHANALEKSHMI MAM & VIKASH SIR <<-- \|n"<<endl;
    gotoxy(17,13);
    cout<<">>>----Project Implemented By-----<<<"<<endl;
    gotoxy(22,15);
    cout<<"CHIRAG SHARMA(LATG10024)"<<endl;
    gotoxy(22,16);
    cout<<"PRAKHAR JAIN(LATG10025)"<<endl;
    gotoxy(22,17);
    cout<<"ARYAN GARG(LATG10046)"<<endl<<endl;
    system("pause");
    system("cls");
    system("color Fc");

    int ch;
    while(ch!=3){

        //      choice for below message

        cout<<"\n\t|-----<Main Menu>-----|";

```

```
cout<<"\n\n";
    cout<<"\t\t 1)Administator of Market\n";
    cout<<"\t\t 2)Customer          \n";
    cout<<"\t\t 3)Exit              \n";

    cout<<"\nEnter Your choice >>>";cin>>ch;
    switch(ch){
    case 1:

        administator();
        break;

    case 2:
        cout<<endl<<endl;
        bpop();
        system("pause");
        buy();
        break;

    case 3:
        cout<<"\n\n\t\t\t\t\tThank You\t\t\t\t\t";
        break;

    }
}
return 0; }
```

CHAPTER 6

CONCLUSION

The conclusions can be taken throughout the development of the Shopping Club Management System with a few suggestions and insights on the achievement of overall objectives, improvements, advantages, and disadvantages of the system. Additionally, suggestions for improving this system in the future are also highlighted.

Overall objective achievement is satisfactory based on the results of the testing of the Shopping Club Management System. All the modules developed in this system work well and achieve the objectives and scope of the project.

Therefore, it is a user friendly system which helps you to purchase products at your own comfort and helps to manage products in the market by administrator.

CHAPTER 7

REFERENCES

- <https://www.geeksforgeeks.org/data-structures/linked-list/>
- <https://www.geeksforgeeks.org/queue-data-structure/>
- <https://www.geeksforgeeks.org/stack-data-structure/>
- <https://www.geeksforgeeks.org/array-data-structure/>