

STOCK MANAGEMENT SYSTEM

PROJECT

TEAM 12 | S3 RSET

INTRODUCTION

Under the Object Oriented Programming Course being offered for S3 CSBS Students of RSET, students have been assigned a mini project to implement.

Teams are to develop a source code in C++ based on UML diagrams. It must include a Command User Interface with tab and arrow keys to select the menu (preferably), File storage wherever necessary, and a Fully functional C++ based application

QUESTION

The stock management system aims at automating the stock management activities in the shop. The shop owner can keep a record of all the items available in the shop.

He has the authority to add, delete and modify the items in the record. This record will be automatically updated on every purchase made.

The shop owner can view the record at any time for stock verification. The system also provides the facility for automatic stock report generation of weekly activities. This system will help to reduce the time and cost of stock management

TEAM MEMBERS

Megha Rajesh

U2109039

Roll No.: 39

Jayasankar C M

U2109029

Roll No.: 29

Mathew Zachariah

U2109038

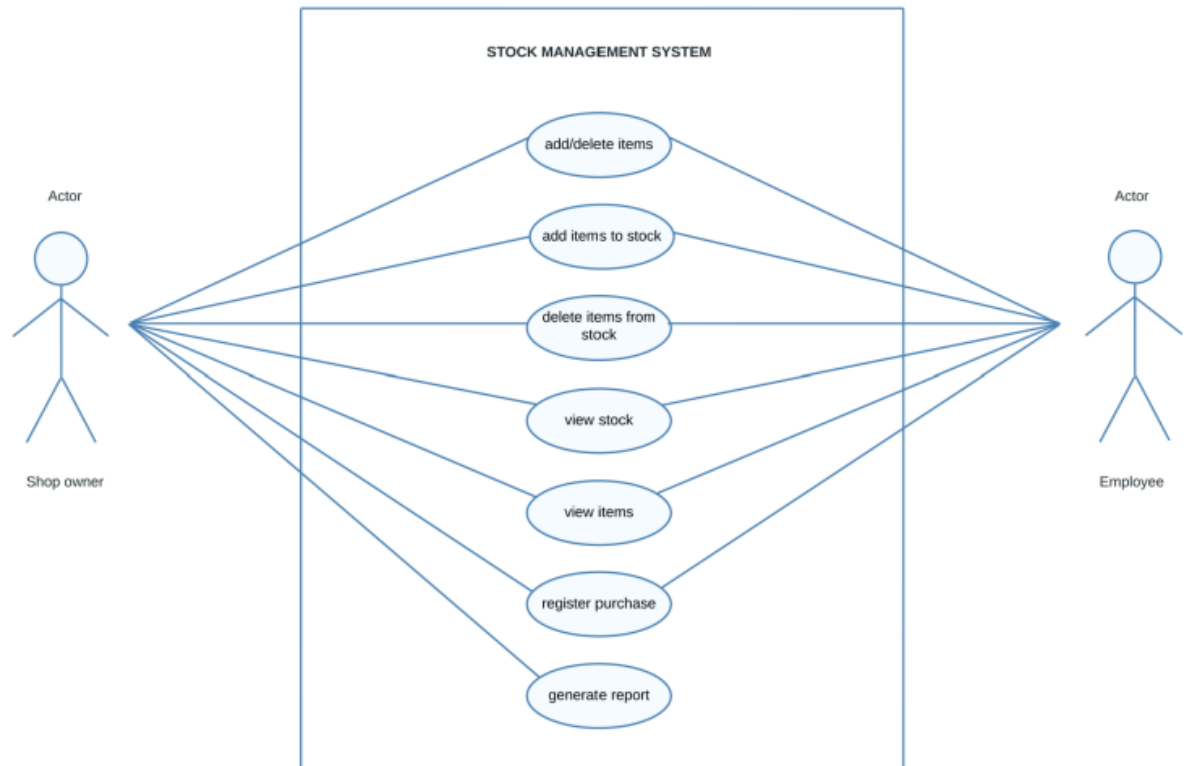
Roll No.: 38

Mohammed Ahsan

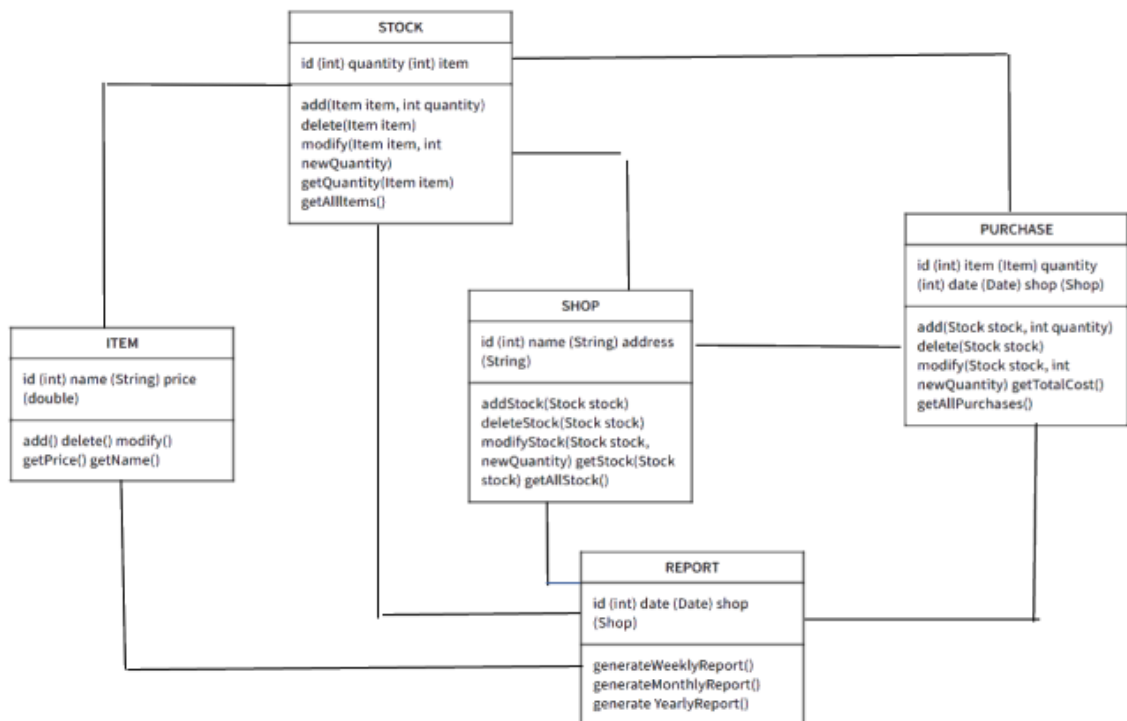
U2109045

Roll No.: 45

USE CASE DIAGRAM

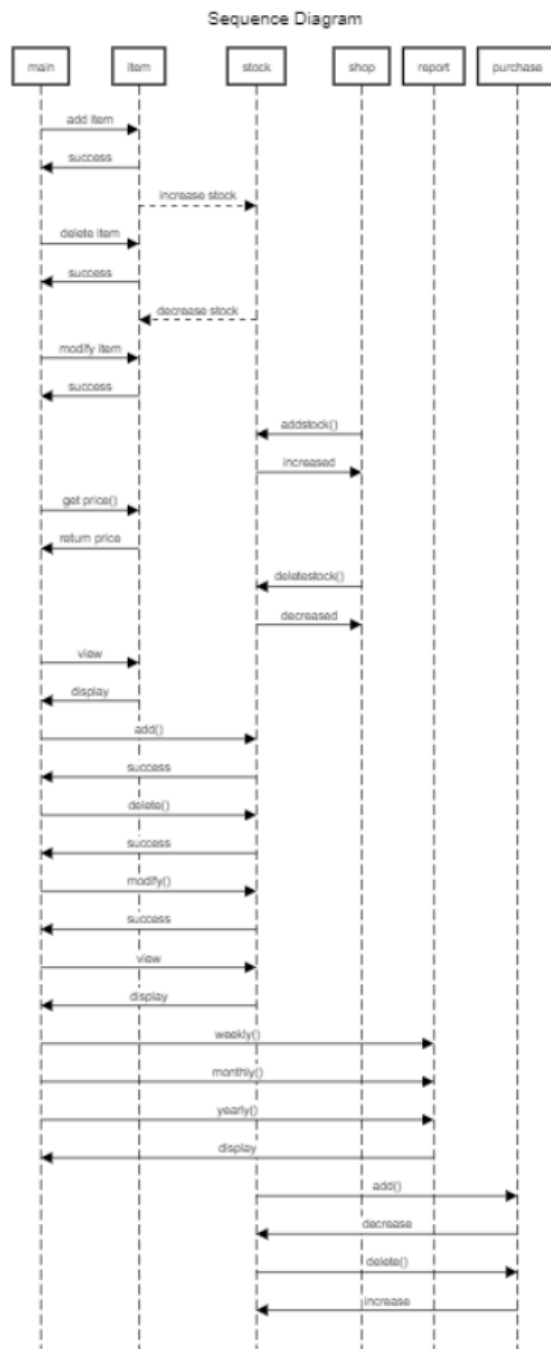


CLASS DIAGRAM



ACTIVITY DIAGRAM

SEQUENCE DIAGRAM



CODE

```
#include <iostream>
#include <string>
#include <stdlib.h>
using namespace std;

class Item{
public:
    int id[100];
    string name[100];
    int quantity[100];
    int price[100];
    string getName(int i) {
        return name[i];
    }
    int getId(int i) {
        return id[i];
    }
    int getQuantity(int i) {
        return quantity[i];
    }
}
```

```

    }
    int getPrice(int i) {
        return price[i];
    }
    void setQuantity(int j,int quant) {
        quantity[j] = quant;
    }
    void setPrice(int j, int pric){
        price[j]= pric;
    }
};

```

```

class Stock: public Item{
public:
    int numItems;
    Stock()
    {
        numItems=0;
    }
    void addItem() {
        cout<<"\nEnter Item ID: ";
        cin>>id[numItems];
        cout<<"\nEnter Item Name: ";
        cin>>name[numItems];
        cout<<"\nEnter Item quantity: ";
        cin>>quantity[numItems];
    }
};

```

```
    cout<<"\nEnter Item Price: ";
    cin>>price[numItems];
    numItems+=1;
}

void updateItem(int uname,int quant,int uprice) {
int flag2=0;
for (int j = 0; j < numItems; j++) {
    if (getIId(j) == uname) {
        setPrice(j,uprice);
        setQuantity(j,quant);
        flag2=1;
        break;
    }
}
if(flag2!=1)
{
    cout<<"Item not found";
}
else
{
    cout<<"Item updated";
}
}

void deleteItem(int dname) {
int flag=0;
for (int j = 0; j < numItems; j++) {
```

```

    if (getId(j) == dname) {
        for(int n=j;n<(numItems-1);n++)
        {
            name[n]=name[n+1];
            id[n]=id[n+1];
            quantity[n]=quantity[n+1];
            price[n]=price[n+1];
        }
        j--;
        numItems--;
        flag=1;
        break;
    }
}
if(flag!=1)
{
    cout<<"Item not found";
}
else
{
    cout<<"Item deleted!"<<endl;
}
}

void displayStock() {
    cout << "\nStock Details:" << endl;
    for (int j = 0; j < numItems; j++) {

```

```

        cout<<endl<< "Id: " << getId(j) << endl;
        cout<< "Name: " << getName(j) << endl;
        cout<< "Quantity: " << getQuantity(j) << endl;
        cout<< "Price: " << getPrice(j) << endl;
    }
}
};

class Purchase:public Stock
{
public:
    int pnum;
    int purchItemid[100];
    int purchQuantity[100];
    int purchPrice[100];
    int purchId[100];
    Purchase()
    {
        pnum=0;
    }
    void purchaseItem(int pItem,int pQnt, Stock &st)
    {
        purchItemid[pnum]=pItem;
        purchQuantity[pnum]=pQnt;
        for(int i=0;i<st.numItems;i++)
        {
            if(st.id[i]==purchItemid[pnum])

```

```

        {
            purchPrice[pnum]=st.price[i];
            st.quantity[i]-=purchQuantity[pnum];
        }
    }

    purchPrice[pnum]=purchPrice[pnum]*purchQuantity[pnum];
    purchId[pnum]=pnum+1;
    pnum++;
    cout<<"Item purchased successfully"<<endl;
}

void printBill(int x)
{
    cout<< "Purchase Details:" << endl<<endl;
    cout<<"Purchase Id: "<<purchId[x]<<endl;
    cout<<"Item Id: "<<purchItemid[x]<<endl;
    cout<<"Quantity: "<<purchQuantity[x]<<endl;
    cout<<"Total price: " <<purchPrice[x]<<endl<<endl;
}

void printrecentBill()
{
    cout<< "Purchase Details:" << endl<<endl;
    cout<<"Purchase Id: "<<purchId[pnum-1]<<endl;
    cout<<"Item Id: "<<purchItemid[pnum-1]<<endl;
    cout<<"Quantity: "<<purchQuantity[pnum-1]<<endl;
    cout<<"Total price: " <<purchPrice[pnum-1]<<endl<<endl;
}

```

```

void generateReport(Stock &st,Purchase &pch)
{
    cout<<"\n\n\t\tREPORT\n\n";
    cout<<"\tSTOCK\n";
    st.displayStock();
    cout<<"\t\tPURCHASES\n\n";
    for(int i=0;i<pnum;i++)
    {
        pch.printBill(i);
    }
}
};

```

```

int main() {
    system("clear");
    int ch1,ch2,ch3;
    string pass;
    Stock st;
    Item it;
    Purchase pch;
    label1:
    while(1)
    {
        system("clear");    //user login page loop
        cout<<endl<<endl<<"\t\tUSER LOGIN"<<endl<<endl;
        cout<<"1. Shop owner login\n2. Employee login\n3. Exit"<<endl<<endl;

```

```

cout<<"Enter your choice:\t";
cin>>ch1;
switch(ch1)
{
case 1:
{
int c1=0;
system("clear");
while(1)//password checking
{
cout<<"Enter password:\t";
cin>>pass;
cout<<endl;
if(pass=="RSET123!")
{
label3:
system("clear");
int c3=0;
while(1)
{
//shop owner ui with report included
cout<<"\n\n\t\tSTOCK MANAGEMENT SYSTEM (SHOP
OWNER)"<<endl;
cout<<"\n1. Add item\n2. Delete item\n3. Update Item\n4. Display
Stock\n5. Manage purchases\n6. Generate report\n7. Exit\n";
cout<<endl<<"Enter your choice:\t";
cin>>ch3;

```



```
switch(ch3)
{
    case 1:{
        st.addItem();
        cout<<"\nItem added!";
        break;
    }
    case 2:{
        int del;
        cout<<"\nEnter item ID to delete:\t";
        cin>>del;
        st.deleteItem(del);
        break;
    }
    case 3:{
        int ups;
        int upi,upp;
        cout<<"\nEnter item ID to update:\t";
        cin>>ups;
        cout<<"\nNew quantity:\t";
        cin>>upi;
        cout<<"\nNew price:\t";
        cin>>upp;
        st.updateItem(ups,upi,upp);
        break;
    }
}
```

```

case 4:{
    st.displayStock();
    break;
}
case 5:{
    system("clear");
    while(1)
    {
        int ch5;
        cout<<"\n\nPURCHASES MENU\n\n";
        cout<<"1. Add purchase\n2. Print bill for last purchase\n3.
Exit"<<endl<<endl;
        cout<<"Enter your choice:\t";
        cin>>ch5;
        switch(ch5)
        {
            case 1:
            {
                int iid,qq;
                cout<<"Enter Item id: ";
                cin>>iid;
                cout<<"Enter quantity: ";
                cin>>qq;
                pch.purchaseItem(iid,qq,st);
                break;
            }

```

```
        case 2:
        {
            pch.printrecentBill();
            break;
        }
        case 3:
        {
            goto label3;
        }
    }
}
case 6:{
    pch.generateReport(st,pch);
    break;
}
case 7:{
    cout<<"\n\n\t\tTHANK YOU"<<endl<<endl<<endl;
    c3=1;
    break;
}
default:{
    cout<<"\nInvalid choice, please try again";
    break;
}
}
```

```

if(c3==1)
{
    goto label1;
}
}
}
else if(pass=="EXIT")
{
    c1=1;
}
else
{
    cout<<"Invalid password, please try again"<<endl;
    cout<<"(Type EXIT to go back)"<<endl<<endl;
}
if(c1==1)
{
    goto label1;
}
}
}
case 2:
{
    label2:
    system("clear");
    int c2=0;

```

```

while(1)
{
    //employee ui without report
    cout<<"\n\n\t\tSTOCK MANAGEMENT SYSTEM
(EMPLOYEE)"<<endl;

    cout<<"\n1. Add item\n2. Delete item\n3. Update Item\n4. Display
Stock\n5. Manage purchases\n6. Exit\n";

    cout<<endl<<"Enter your choice:\t";

    cin>>ch2;
    switch(ch2)
    {
        case 1:{
            st.addItem();
            cout<<"\nItem added!";
            break;
        }
        case 2:{
            int del;
            cout<<"\nEnter item ID to delete:\t";
            cin>>del;
            st.deleteItem(del);
            break;
        }
        case 3:{
            int ups;
            int upi,upp;
            cout<<"\nEnter item ID to update:\t";

```

```

        cin>>ups;
        cout<<"\nNew quantity:\t";
        cin>>upi;
        cout<<"\nNew price:\t";
        cin>>upp;
        st.updateItem(ups,upi,upp);
        break;
    }
    case 4:{
        st.displayStock();
        break;
    }
    case 5:{
        system("clear");
        while(1)
        {
            int ch5;

            cout<<"\n\nPURCHASES MENU\n\n";
            cout<<"1. Add purchase\n2. Print bill for last purchase\n3.
Exit"<<endl<<endl;

            cout<<"Enter your choice:\t";
            cin>>ch5;
            switch(ch5)
            {
                case 1:
                {

```

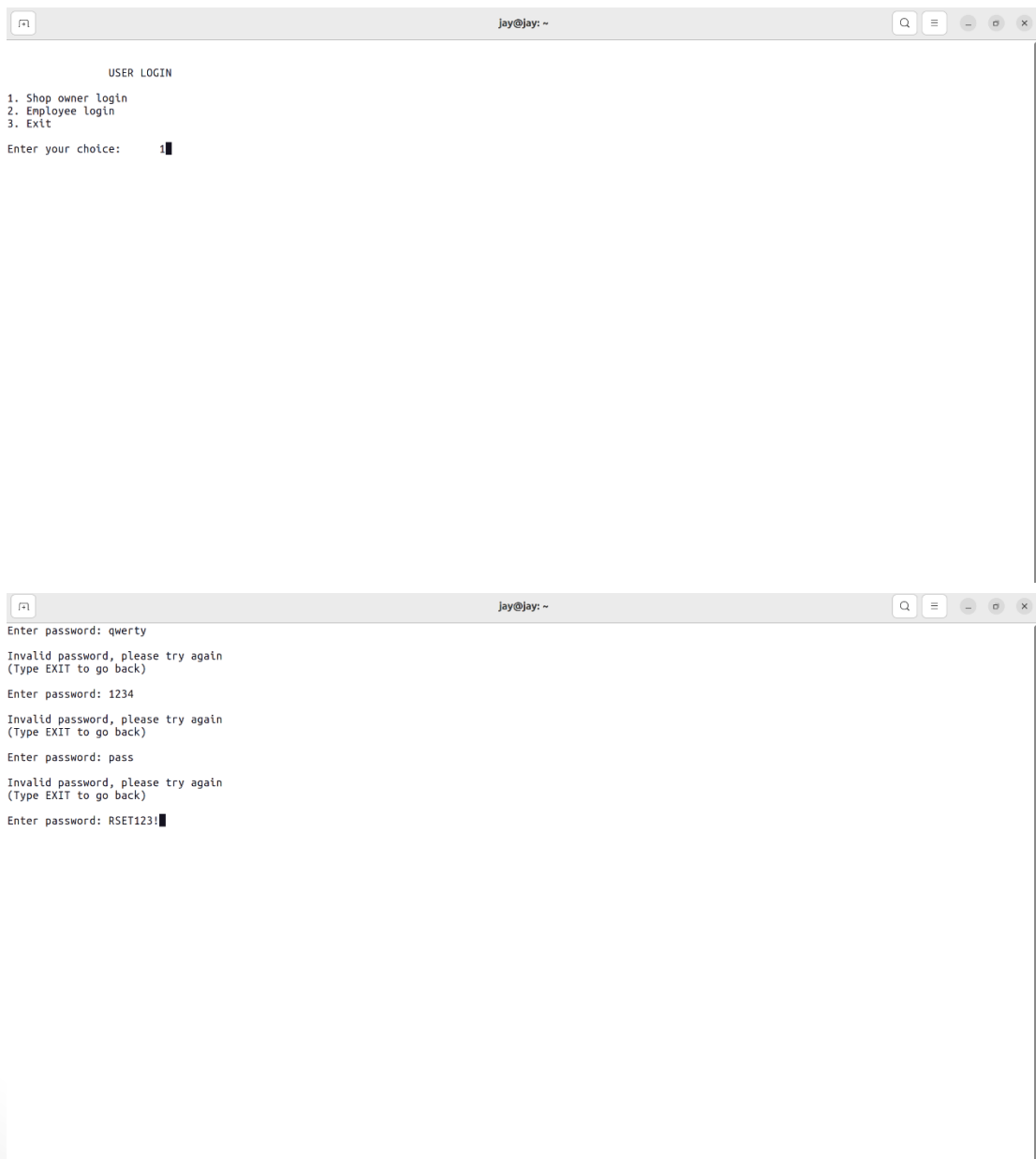
```
int iid,qq;
cout<<"Enter Item id: ";
cin>>iid;
cout<<"Enter quantity: ";
cin>>qq;
pch.purchaseItem(iid,qq,st);
break;
}
case 2:
{
pch.printrecentBill();
break;
}
case 3:
{
goto label2;
}
}
}
}
case 6:{
cout<<"\n\n\t\tTHANK YOU"<<endl<<endl<<endl;
c2=1;
break;
}
default:{
```

```

        cout<<"\nInvalid choice, please try again";
        break;
    }
}
if(c2==1)
{
    goto label1;
}
}
}
case 3:{
    cout<<endl<<endl<<"\t\t\tTHANK YOU"<<endl<<endl<<endl;
    return(0);
}
default:
{
    cout<<"Invalid choice, please try again";
    break;
}
}
}
}

```


TEST CASES AND RESULT



The image shows two screenshots of a terminal window titled 'jay@jay: ~'. The first screenshot displays the 'USER LOGIN' menu with three options: '1. Shop owner login', '2. Enployee login', and '3. Exit'. Below the menu, it says 'Enter your choice:' followed by a cursor. The second screenshot shows a series of password attempts. It starts with 'Enter password: qwerty', followed by an error message 'Invalid password, please try again (Type EXIT to go back)'. This is repeated for '1234' and 'pass'. The final attempt is 'RSET123!' followed by a cursor.

```
jay@jay: ~  
USER LOGIN  
1. Shop owner login  
2. Enployee login  
3. Exit  
Enter your choice: █  
  
jay@jay: ~  
Enter password: qwerty  
Invalid password, please try again  
(Type EXIT to go back)  
Enter password: 1234  
Invalid password, please try again  
(Type EXIT to go back)  
Enter password: pass  
Invalid password, please try again  
(Type EXIT to go back)  
Enter password: RSET123!█
```

```
jay@jay: ~  
  
STOCK MANAGEMENT SYSTEM (SHOP OWNER)  
1. Add item  
2. Delete item  
3. Update Item  
4. Display Stock  
5. Manage purchases  
6. Generate report  
7. Exit  
Enter your choice: 1  
Enter Item ID: 101  
Enter Item Name: soap  
Enter Item quantity: 120  
Enter Item Price: 60  
Item added!  
  
STOCK MANAGEMENT SYSTEM (SHOP OWNER)  
1. Add item  
2. Delete item  
3. Update Item  
4. Display Stock  
5. Manage purchases  
6. Generate report  
7. Exit  
Enter your choice: 1  
Enter Item ID: 102  
Enter Item Name: pen  
Enter Item quantity: 100  
Enter Item Price: 20  
Item added!  
  
STOCK MANAGEMENT SYSTEM (SHOP OWNER)  
1. Add item  
2. Delete item  
3. Update Item  
4. Display Stock  
5. Manage purchases  
6. Generate report  
7. Exit  
Enter your choice: 1  
Enter Item ID: 103  
Enter Item Name: book  
Enter Item quantity: 70  
Enter Item Price: 50  
Item added!  
  
STOCK MANAGEMENT SYSTEM (SHOP OWNER)  
1. Add item  
2. Delete item  
3. Update Item  
4. Display Stock  
5. Manage purchases  
6. Generate report  
7. Exit  
Enter your choice: 1  
Enter Item ID: 104  
Enter Item Name: bottle
```

```
jay@jay: ~  
Enter Item ID: 104  
Enter Item Name: bottle  
Enter Item quantity: 80  
Enter Item Price: 100  
Item added!  
  
STOCK MANAGEMENT SYSTEM (SHOP OWNER)  
  
1. Add item  
2. Delete item  
3. Update Item  
4. Display Stock  
5. Manage purchases  
6. Generate report  
7. Exit  
  
Enter your choice: 4  
  
Stock Details:  
  
Id: 101  
Name: soap  
Quantity: 120  
Price: 60  
  
Id: 102  
Name: pen  
Quantity: 100  
Price: 20  
  
Id: 103  
Name: book  
Quantity: 70  
Price: 50  
  
Id: 104  
Name: bottle  
Quantity: 80  
Price: 100  
  
jay@jay: ~  
Name: bottle  
Quantity: 80  
Price: 100  
  
STOCK MANAGEMENT SYSTEM (SHOP OWNER)  
  
1. Add item  
2. Delete item  
3. Update Item  
4. Display Stock  
5. Manage purchases  
6. Generate report  
7. Exit  
  
Enter your choice: 3  
  
Enter item ID to update: 104  
  
New quantity: 80  
New price: 100  
Item updated  
  
STOCK MANAGEMENT SYSTEM (SHOP OWNER)  
  
1. Add item  
2. Delete item  
3. Update Item  
4. Display Stock  
5. Manage purchases  
6. Generate report  
7. Exit  
  
Enter your choice: 3  
  
Enter item ID to update: 102  
  
New quantity: 80  
New price: 25  
Item updated
```

```
jay@jay: ~
New price: 25
Item updated

STOCK MANAGEMENT SYSTEM (SHOP OWNER)

1. Add item
2. Delete item
3. Update Item
4. Display Stock
5. Manage purchases
6. Generate report
7. Exit

Enter your choice: 2

Enter item ID to delete: 104
Item deleted!

STOCK MANAGEMENT SYSTEM (SHOP OWNER)

1. Add item
2. Delete item
3. Update Item
4. Display Stock
5. Manage purchases
6. Generate report
7. Exit

Enter your choice: 4

Stock Details:

Id: 101
Name: soap
Quantity: 120
Price: 60

Id: 102
Name: pen
Quantity: 80
Price: 25

jay@jay: ~

STOCK MANAGEMENT SYSTEM (SHOP OWNER)

1. Add item
2. Delete item
3. Update Item
4. Display Stock
5. Manage purchases
6. Generate report
7. Exit

Enter your choice: 4

Stock Details:

Id: 101
Name: soap
Quantity: 120
Price: 60

Id: 102
Name: pen
Quantity: 80
Price: 25

Id: 103
Name: book
Quantity: 70
Price: 50

STOCK MANAGEMENT SYSTEM (SHOP OWNER)

1. Add item
2. Delete item
3. Update Item
4. Display Stock
5. Manage purchases
6. Generate report
7. Exit

Enter your choice: 5
```

```
Jay@Jay: ~  
  
PURCHASES MENU  
1. Add purchase  
2. Print bill for last purchase  
3. Exit  
  
Enter your choice: 1  
Enter Item id: 101  
Enter quantity: 4  
Item purchased successfully  
  
PURCHASES MENU  
1. Add purchase  
2. Print bill for last purchase  
3. Exit  
  
Enter your choice: 1  
Enter Item id: 102  
Enter quantity: 12  
Item purchased successfully  
  
PURCHASES MENU  
1. Add purchase  
2. Print bill for last purchase  
3. Exit  
  
Enter your choice: 2  
Purchase Details:  
  
Purchase Id: 2  
Item Id: 102  
Quantity: 12  
Total price: 300  
  
PURCHASES MENU  
3. Exit  
  
Enter your choice: 1  
Enter Item id: 101  
Enter quantity: 4  
Item purchased successfully  
  
PURCHASES MENU  
1. Add purchase  
2. Print bill for last purchase  
3. Exit  
  
Enter your choice: 1  
Enter Item id: 102  
Enter quantity: 12  
Item purchased successfully  
  
PURCHASES MENU  
1. Add purchase  
2. Print bill for last purchase  
3. Exit  
  
Enter your choice: 2  
Purchase Details:  
  
Purchase Id: 2  
Item Id: 102  
Quantity: 12  
Total price: 300  
  
PURCHASES MENU  
1. Add purchase  
2. Print bill for last purchase  
3. Exit  
  
Enter your choice:
```

```
jay@jay: ~  
  
STOCK MANAGEMENT SYSTEM (SHOP OWNER)  
  
1. Add item  
2. Delete item  
3. Update Item  
4. Display Stock  
5. Manage purchases  
6. Generate report  
7. Exit  
  
Enter your choice: 6  
  
REPORT  
  
STOCK  
  
Stock Details:  
  
Id: 101  
Name: soap  
Quantity: 116  
Price: 60  
  
Id: 102  
Name: pen  
Quantity: 68  
Price: 25  
  
Id: 103  
Name: book  
Quantity: 70  
Price: 50  
  
PURCHASES  
  
Purchase Details:  
  
Purchase Id: 1  
Item Id: 101  
Quantity: 4  
Total price: 240  
  
Id: 101  
Name: soap  
Quantity: 116  
Price: 60  
  
Id: 102  
Name: pen  
Quantity: 68  
Price: 25  
  
Id: 103  
Name: book  
Quantity: 70  
Price: 50  
  
PURCHASES  
  
Purchase Details:  
  
Purchase Id: 1  
Item Id: 101  
Quantity: 4  
Total price: 240  
  
Purchase Details:  
  
Purchase Id: 2  
Item Id: 102  
Quantity: 12  
Total price: 300  
  
STOCK MANAGEMENT SYSTEM (SHOP OWNER)  
  
1. Add item  
2. Delete item  
3. Update Item  
4. Display Stock  
5. Manage purchases  
6. Generate report  
7. Exit  
  
Enter your choice: 7
```

