VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANASANGAMA, BELAGAVI – 590018

Mini Project Report

Stock Management System

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering

In

Information Science and Engineering

Submitted by

R Dhanesh

Anuj S Rakaraddi

RV Institute of Technology and Management®

(Affiliated to VTU, Belagavi)

JP Nagar, Bengaluru – 560076

Information Science and Engineering

Information Science and Engineering

CERTIFICATE

Certified that the Mini project entitled "**Stock Management System"** carried out by **R Dhanesh Anuj S Rakaraddi** are bonafied students of I Semester B.E, **RV Institute of Technology and Management** in partial fulfilment for the Bachelor of Engineering in **Information Science and Engineering**, of the **Visvesvaraya Technological University,** Belagavi, during the academic year 2020 – 2021. The Mini projectreport has been approved as it satisfies the academic requirements in respect of Data Structures and Applications.

Dr. Savitha G & Dr. Deepak N A

Associate Professor Associate Professor

Dept.CSE

RVITM, Bengaluru – 560076

Index

Chapter 1.

Introduction to Stock Management System(SMS)

The Existing Stock Management System Has Been A Very Non Efficient One, It is time consuming and non effective, this makes it non economical for a organization to use it as it uses lots of manual labour and unorganized paper work.

So,The Stock Management System (SMS) refers to the system and processes to manage the stock of organization with the involvement of Technology . 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 report daily or weekly based. This project is categorize individual aspects for the sales and stock management system. In this system we are solving different problem affecting to direct sales management and purchase management.
Stock Management System is important to ensure quality control in businesses that handle transactions resolving 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 on important means of automatically tracking large shipment. An automated Stock Management System helps to minimize the errors while recording the stock.

Chapter 2

Algorithm

Start:

1) Import Java util.* ;

This imports the scanning, and the HashMap.

2) Data Structure

Data Structure used here is HashMap.

HashMap:

It provides the basic implementation of the Map interface of Java. It stores the data in (Key, Value) pairs, and you can access them by an index of another type (e.g. an Integer). One object is used as a key (index) to another object (value). If you try to insert the duplicate key, it will replace the element of the corresponding key.

Part 1: Storing data.

► Two HashMap are used in this program.

► One HashMap is for the stocks(hm1) and the other is for the costs of the items(hm2).

► The HashMap used here are of the type of String to Integer.

Where the user stores the item name in the key of the

HashMap and stores the cost or stock in the value of the HashMap

► Initializing the HashMap.

Here hm1 takes the stock values, and hm2 takes the price of the items.

```
HashMap<String,Integer> hm1=new
HashMap<String,Integer>();

        HashMap<String,Integer> hm2=new
HashMap<String,Integer>();
```

► To add the item and the stocks,

```
System.out.print("item:");
String item=scan.next();
System.out.print("stock:");
int stock=scan.nextInt();
hm1.put(item,stock);
```

► To add the price the code is

```
System.out.print("item:");
String item=scan.next();
System.out.print("cost:");
int cost=scan.nextInt();
hm2.put(item,cost);
```

► If user wants to exit from providing information, user needs to Enter (-1).

Part 2: Billing.

While billing, the price of the item is known by hm2, and the stock value is deducted from hm1.

```
con_exit
int sum=0;
while(con_exit==1)
{
System.out.print("item:");
Stringcheck=scanext();
System.out.print("x");
int times=scan.nextInt(); int
temp=hm1.get(check); temp=temp-times;
hm1.remove(check);
hm1.put(check,temp);
int bill=hm2.get(check); bill=bill*times;
sum=sum+bill;System.out.println("current
total:"+sum); System.out.println("1 to
continue and -1 to exit
con_exit=scan.nextInt();
}
```

If user wants to exit from providing information, user needs to Enter (-1).

Part 3: Printing stock value.

►If the user wants to know the stocks at the moment, he needs to Enter “stock”.

- The code to print the stocks is

```
System.out.println("your current stock is "+hm1);
```

Part 4: Assembling part 1, part 2 and part 3 in main function.

►An infinite while loop is used, so that the user can get to whatever parts he needs after end of the loop.

►Inside the Infinite while loop the user will be asked what does he want to do,

```
System.out.println("TO DO?");
```

→If user wants to provide information, user needs to enter ‘add’.

→If user wants to go to billing, user needs to enter ‘bill’.

→If user wants to know the stock value, user needs to enter ‘stocks’.

►The input will be scanned by the system.

```
String to_do=scan.next();
```

►If the input is ‘add’, System will take the user to provide information i.e. part 1.

```
if(to_do.equals("add"))
        {
            //part 1;
        }
```

►If the input is ‘billing’, System will take the user to billing i.e. part 2.

```
if(to_do.equals("bill"))
{
```

//part 2;

}

►If the input is ‘stocks’, System will print the stocks i.e. part 3.

if(to_do.equals("stock"))

//part3;

Chapter 3:

Implementation details/code.

```
Import java.util.*;
public class
Stocks
{
        public static void main(String[] args

");
```

```
Scanner scan= new Scanner      (System.in); int con_exit=1;
 HashMap<String,Integer> hm1=new HashMap<String,Integer>();
 HashMap<String,Integer> hm2=new HashMap<String,Integer>();
 System.out.println("to provide INFORMATION Enter 'add'");
 System.out.println("to bill Enter 'bill'");
 System.out.println("to know the stocks type 'stock'");
 System.out.println("NOTE: first you need to provide INFORMATION

 while(1>0)
   {
System.out.println("TO DO?"); String
to_do=scan.next();
if(to_do.equals("add"))
                {

                   System.out.println("PROVIDE YOUR INFORMATION");
                   while(con_exit==1)
                     {
                        System.out.print("item:");
```

```
String item=scan.next();
System.out.print("stock:");
        int stock=scan.nextInt();
                        System.out.print("cost:
                        "); int
                        cost=scan.nextInt();
                        hm1.put(item,stock);
                        hm2.put(item,cost);
                        System.out.println("enter 1 to continue and -1
                        to exit"); con_exit=scan.nextInt();
                    }
                }
                  System.out.println();

                  if(to_do.equals("bill"))

                  {

                  System.out.println("BILLING");
                  con_exit=1;
                  int sum=0;
                  while(con_exit==
                  1)
                   {
System.out.print("item:");
String                  check=scan.next();
System.out.print("x");
int times=scan.nextInt();
int temp=hm1.get(check);
temp=temp-times;
```

```
hm1.remove(check);
hm1.put(check,temp);
int bill=hm2.get(check);
bill=bill*times;
sum=sum+bill;
System.out.println("current total:"+sum);
System.out.println("enter 1 to continue and –1 to exit");
con_exit=scan.nextInt();
                    }

                   }
                   System.out.println();
              if(to_do.equals("stock"))
                {
                          System.out.println("your current stock is "+hm1);

}
         }
 }
```

Chapter 4:

Experimental results/Snapshots

```
to privide INFORMATION Enter 'add'
to bill Enter 'bill'
to know the stocks type 'stock'
NOTE: first you need to provide INFORMATION
TO DO?
```

```
to privide INFORMATION Enter 'add'
to bill Enter 'bill'
to know the stocks type 'stock'
NOTE: first you need to provide INFORMATION
TO DO?
add
PROVIDE YOUR INFORMATION
item:mango
stock:300
cost:80
enter 1 to continue and -1 to exit
1
item:apple
stock:250
cost:100
enter 1 to continue and -1 to exit
-1


TO DO?
```

```
TO DO?
bill


BILLING
item:mango
x30
current total:2400
enter 1 to continue and -1 to exit
1
item:apple
x12
current total:3600
enter 1 to continue and -1 to exit
-1
TO DO?
```

```
TO DO?
stock


your current stock is {apple=238, mango=270}
```