Computer Science Project

SUPERMARKET BILLING SYSTEM

By: Aditya Mohana Sivaraj
Class: XII A8
Roll Number: 04

CONTENTS

PROBLEM DEFINITON

DEMRIS, a departmental store, has provided services to customers over the past 20 years. Ever since the billing, Maintenance, and Purchase has been manual and the number of customers per day has increased tremendously and there has been a decision from the management to make this automated. The department performs the following activities:

- Billing for the customer.
- Maintenance of stock.
- Purchase of stock.

Problems of the existing system with the manual process are:

- Billing for the customer is quite time consuming.
- Users have to stand in a queue for a long time to collect the products after billing and cash credit.
- Maintenance of stock and placing order for the new purchase is time consuming.

Proposed System for Computerization

The management of DEMRIS has decided to computerize the above-mentioned functions of billing as well as maintenance and purchase.

Process Steps

1) Open a new bill for the customer.
2) New Entry form for purchasing a particular product.
3) Generating reports for stock value, and quantity of products on hand.

Checks and validations

1) Number allocated to each product must be unique.
2) Quantity on hand must always be less than or equal to the stock value.
3) When billing is done, the sold out quantity should be updated automatically.
4) The purchase happens through a vendor application, which is always successful.

PROBLEM ANALYSIS

Two core databases are required here, one for the product list and the other for maintaining dealer information. These are created and maintained through binary files.

Binary files used

1) *SHOP.DAT* – Stores all the details of all the products.
2) *BILL.DAT* – Stores the bill number for every transaction.
3) *DEALER.DAT* – Store the details of all the dealers.

The various segments (classes and functions) of the program are elucidated below.

Class "product"
This class is used to create a product and store its details - Product number, Name, Price, Discount, Stock In, Stock out, and remaining stock. There is also a member called *qq*, of integer type which takes the value of Stock In (default stock) and uses it for manipulation.

It has these member functions:

void create_product() is used to input the details for a newly created object of type product.

void show_product() is used to print the details of a given product.

void s_mod(int, int) and *void qq_mod(int)* are used to update the value of that product's stock after every purchase, i.e. bill generation.

void s_set() is used to replenish the stock of some product to its base value after a purchase for that product is made.

The accessor functions for the data members are *int retpno()*, *float retprice()*, *char* retname()*, *int retdis()*, *int rets_in()*, *int rets_bal()*, *int rets_out()*, *int retqq()*

Class “bill”

This class is used to store the bill number for every purchase and its object is appended to the binary file BILL.DAT for dynamic, sequential assignment of the bill number when a bill is made.

It has these member functions:

void in(int) is used to change, i.e. increment the bill number by one after every transaction. This value is appended to the file.

void retbno() is an accessor function that returns the bill number.

Class “dealer”

Similar in design to the class product, this class is used to create an object of type dealer and store their information – Dealer number, Name, Address, Mobile number, Phone number.

It has these member functions:

void create_dealer() is used to input the details for a newly created object of type dealer.

void show_dealer() is used to print the details of a given dealer.

The accessor functions for the data members are *char* retadd()*, *char* retdname()*, *char* retphno()*, *char* mobno()*, *int retdno()*

User defined functions outside the class

void write_product() is used to write records, i.e different product details onto the file, SHOP.DAT

void display_all() is used to show details of all the products.

void display_sp(int n) to show details of a particular product. This function is called in case a query is made.

void modify_product() is used to modify details of a product.

void delete_product() is used to delete a product from the inventory list.

void menu() is used to display all products price list.

void place_order() is used to place and order for products and generate a bill.

void intro() shows Main menu options for the user

void admin_menu() is a Menu function to show list of admin operations.

void deal_menu() is a Menu function to show dealer operations.

void write_dealer() writes records, i.e different dealer details onto the file, DEALER.DAT

void display_all_d() shows details of all the dealers.

void display_sd(int n) is used to show details of a particular dealer.

void delete_product() is used to remove a dealer from the file.

Report Generation

void s_report() makes the stock report.

FUTURE ENHANCEMENTS

1) A data encryption system may be incorporated to keep data encrypted as an option to add security to the project.
2) Usage of dynamic data structures such as linked lists for the backbone of the billing module to make the run program faster and save memory.
3) Implementation of “auto fill” feature in areas like Bill generation and Purchase Order wherein product name/dealer name, and details appear as soon as the user enters the product number/dealer number.
This enhances the user interface further and makes the program handy.
4) Stringent checks to be placed at every point of data entry into database, to prevent redundancy.
5) Addition of customer grievance redressal service, i.e., Customer will now have the provision to return back faulty goods within 15 days of purchase, if it is not in tampered condition. To facilitate this, a bill tracking system using bill number, as a key will be made.

HARDWARE REQUIREMENTS

- 10MB free space on c:\TC\BIN
- 256MB cache memory
- 500 MHz Pentium Celeron Processor or better
- CRT/LCD monitor
- Microsoft Basic Display Driver

SOFTWARE REQUIREMENTS

- Windows NT/XP/7
- Turbo C++ 3.0

SOURCE CODE

```
//*****************************************
//DEMRIS Departmental Store Automated Billing
System Project.
//*****************************************
#include<fstream.h>
#include<process.h>
#include<stdio.h>
#include<conio.h>
//****************
// Class "product"
//****************
class product
{
int pno,s_in,s_bal,s_out,qq;
char name[50];
float price,qty,tax,dis;
public:
void create_product()
{
cout<<"\nPlease Enter The Product No. of The Product:
";
cin>>pno;
cout<<"\n\nPlease Enter The Name of The Product: ";
gets(name);
cout<<"\nPlease Enter The Price of The Product: ";
cin>>price;
cout<<"\nPlease Enter The Discount (%): ";
cin>>dis;
cout<<"\nPlease Enter The Basic Stock value: ";
cin>>s_in;qq=s_in;s_out=0;
```

```
}
void s_mod(int a,int b)
{
s_bal=a;
s_out+=b;
}
void qq_mod(int z)
{qq=z;}
void s_set()
{qq=s_in;}
void show_product()
{
cout<<"\nThe Product No. of The Product : "<<pno;
cout<<"\nThe Name of The Product : ";
puts(name);
cout<<"\nThe Price of The Product : "<<price;
cout<<"\nDiscount : "<<dis;
cout<<"\nBasic Stock : "<<s_in;
cout<<"\nStock Out: "<<s_out;
cout<<"\nRem : "<<s_bal;
}
int retpno()
{return pno;}
float retprice()
{return price;}
char* retname()
{return name;}
int retdis()
{return dis;}
int rets_in()
{return s_in;}
int rets_out()
{return s_out;}
```

```
int rets_bal()
{return s_bal;}
int retqq()
{return qq;}
}; //class ends here
//************
//class "bill"
//************
class bill
{
int bno;
public:
void in(int q)
{bno=q;}
int retbno()
{return bno;}
}; //class ends here
//**************
// Class "dealer"
//**************
class dealer
{
int dno;
char add[90],dname[40],phno[11],mobno[10];
public:
void create_dealer()
{
cout<<"\nPlease Enter The Dealer No.: ";
cin>>dno;
cout<<"\n\nPlease Enter The Name of The Dealer: ";
gets(dname);
cout<<"\nPlease Enter The Address of The Dealer: ";
gets(add);
```

```
cout<<"\nPlease Enter The Phone No.: ";
gets(phno);
cout<<"\nPlease Enter The Mobile Number: ";
gets(mobno);
}
void show_dealer()
{
cout<<"\nThe Dealer No.: "<<dno;
cout<<"\nThe Name of The Dealer : ";
puts(dname);
cout<<"\nAddress : "<<add;
cout<<"\nPhone Number: "<<phno;
cout<<"\nMobile Number: "<<mobno;
}
char* retadd()
{return add;}
char* retdname()
{return dname;}
char* retphno()
{return phno;}
char* retmobno()
{return mobno;}
int retdno()
{return dno;}
};//class ends here.

//*****************************************
// global declaration for stream object,product,dealer and
bill object
//*****************************************
fstream fp,fb,fd;
product pr;
bill b1;
```

```
dealer dr;
//***********************************
// function to write in Dealer file
//***********************************
void write_dealer()
{
fd.open("Dealer.dat",ios::out|ios::app);
dr.create_dealer();
fd.write((char*)&dr,sizeof(dealer));
fd.close();
cout<<"\n\nThe Dealer Has Been Added ";
getch();
}
//***********************************
// function to write in Product file
//***********************************
void write_product()
{
fp.open("Shop.dat",ios::out|ios::app);
pr.create_product();
fp.write((char*)&pr,sizeof(product));
fp.close();
cout<<"\n\nThe Product Has Been Created ";
getch();
}
//***********************************************
// function to read all records from Dealer file
//***********************************************
void display_all_d()
{
clrscr();
cout<<"\n\n\n\t\tDISPLAY ALL DEALERS !!!\n\n";
fd.open("Dealer.dat",ios::in);
```

```
while(fd.read((char*)&dr,sizeof(dr)))
{
dr.show_dealer();
cout<<"\n\n================================
====\n";
getch();
}
fd.close();
getch();
}
//*************************************
// function to read all records from file
//*************************************
void display_all()
{
clrscr();
cout<<"\n\n\n\t\tDISPLAY ALL RECORD !!!\n\n";
fp.open("Shop.dat",ios::in);
while(fp.read((char*)&pr,sizeof(product)))
{
pr.show_product();
cout<<"\n\n================================
====\n";
getch();
}
fp.close();
getch();
}
//*****************************************
// function to read specific record from file
//*****************************************
void display_sp(int n)
{
```

```
int flag=0;
fp.open("Shop.dat",ios::in);
while(fp.read((char*)&pr,sizeof(product)))
{
if(pr.retpno()==n)
{
clrscr();
pr.show_product();
flag=1;
}
}
fp.close();
if(flag==0)
cout<<"\n\nrecord not exist";
getch();
}
//***************************************************
// function to read specific record from Dealer file
//***************************************************
void display_sd(int n)
{
int flag=0;
fd.open("Dealer.dat",ios::in);
while(fd.read((char*)&dr,sizeof(dealer)))
{
if(dr.retdno()==n)
{
clrscr();
dr.show_dealer();
flag=1;
}
}
fd.close();
```

```
if(flag==0)
cout<<"\n\nrecord not existing";
getch();
}
//***************************************
// function to modify record of file
//***************************************
void modify_product()
{
int no,found=0;
clrscr();
cout<<"\n\n\tTo Modify ";
cout<<"\n\n\tPlease Enter The Product No. of The
Product:";
cin>>no;
fp.open("Shop.dat",ios::in|ios::out);
while(fp.read((char*)&pr,sizeof(product)) &&
found==0) ;
{
if(pr.retpno()==no)
{
pr.show_product();
cout<<"\nPlease Enter The New Details of
Product"<<endl;
pr.create_product();
int pos=(fp.tellp()-sizeof(pr));
fp.seekp(pos,ios::beg);
fp.write((char*)&pr,sizeof(product));
cout<<"\n\n\t Record Updated";
found=1;
}
}
fp.close();
```

```
if(found==0)
cout<<"\n\n Record Not Found ";
getch();
}
//***************************************
// function to delete record of Dealer file
//***************************************
void delete_dealer()
{
int no;
clrscr();
cout<<"\n\n\n\tDelete Record";
cout<<"\n\nPlease Enter no. of The Dealer you want to
delete: ";
cin>>no;
fd.open("Dealer.dat",ios::in|ios::out);
fstream fd2;
fd2.open("Temp1.dat",ios::out);
fd.seekg(0,ios::beg);
while(fd.read((char*)&dr,sizeof(dealer)))
{
if(dr.retdno()!=no)
{
fd2.write((char*)&dr,sizeof(dealer));
}
}
fd2.close();
fd.close();
remove("Dealer.dat");
rename("Temp1.dat","Dealer.dat");
cout<<"\n\n\tRecord Deleted ..";
getch();
}
```

```
//***********************************
// function to delete record of file
//***********************************
void delete_product()
{
int no;
clrscr();
cout<<"\n\n\n\tDelete Record";
cout<<"\n\nPlease Enter The product no. of The Product
You Want To Delete: ";
cin>>no;
fp.open("Shop.dat",ios::in|ios::out);
fstream fp2;
fp2.open("Temp.dat",ios::out);
fp.seekg(0,ios::beg);
while(fp.read((char*)&pr,sizeof(product)))
{
if(pr.retpno()!=no)
{
fp2.write((char*)&pr,sizeof(product));
}
}
fp2.close();
fp.close();
remove("Shop.dat");
rename("Temp.dat","Shop.dat");
cout<<"\n\n\tRecord Deleted ..";
getch();
}
//*********************************************
// function to display all products price list
//*********************************************
```

```
void menu()
{
clrscr();
fp.open("Shop.dat",ios::in);
if(!fp)
{
cout<<"ERROR!!! FILE COULD NOT BE OPEN\n\n\n
Go To Admin Menu to create File";
cout<<"\n\n\n Program is closing ....";
getch();
exit(0);
}
cout<<"\n\n\t\tProduct MENU\n\n";
cout<<"==========================================
==========================\n";
cout<<"P.NO.\t\tNAME\tPRICE\tSTOCK ON HANDn";
cout<<"==========================================
==========================\n";
while(fp.read((char*)&pr,sizeof(product)))
{
cout<<pr.retpno()<<"\t\t"<<pr.retname()<<"\t\t"<<pr.retp
rice()<<"\t"<<pr.rets_bal()<<endl;
}
fp.close();
}

//*****************************************
// function to place order and generating bill for Products
//*****************************************
void place_order()
{
int p,order_arr[50],quan[50],c=0;
long pos;
```

```
float amt,damt,total=0;
char ch='Y';
menu();
cout<<"\n================================";
cout<<"\n PLACE YOUR ORDER";
cout<<"\n================================\n";
do{
cout<<"\n\nEnter The Product No. Of The Product : ";
cin>>order_arr[c];
cout<<"\nQuantity in number : ";
cin>>quan[c];
c++;
cout<<"\nDo You Want To Order Another Product ?
(y/n)";
cin>>ch;
}while(ch=='y' ||ch=='Y');
cout<<"\n\nThank You For Placing The Order";
getch();
clrscr();
fb.open("bill.dat",ios::binary|ios::in);
if(!fb)
{
cout<<"File cannot open";
getch();
exit(1);
}
while(!fb.eof())
{
fb.read((char*)&b1,sizeof(bill));
}
p=b1.retbno();
p++;
b1.in(p);
```

```
fb.close();
ofstream fb1;
fb1.open("bill.dat",ios::binary||ios::app);
fb1.write((char*)&b1,sizeof(bill));
fb1.close();
cout<<"\n BILL NUMBER: "<<b1.retbno();
cout<<"\n\n**********************************INV
OICE**************************\n";
cout<<"\nPr No.\tPr Name  \tQuantity\tPrice \tAmount
\tAmount after discount\n";
for(int x=0;x<=c;x++)
{
fp.open("Shop.dat",ios::in|ios::out|ios::binary);
fp.read((char*)&pr,sizeof(product));
while(!fp.eof())
{
if(pr.retpno()==order_arr[x])
{
pr.qq_mod((pr.retqq()-quan[x]));
pr.s_mod(pr.retqq(),quan[x]);
amt=pr.retprice()*quan[x];
damt=amt-(amt*pr.retdis()/100);
cout<<"\n"<<order_arr[x]<<"\t"<<pr.retname()
<<"\t"<<quan[x]<<"\t\t"<<pr.retprice()<<"\t"<<amt<<"\t
\t"<<damt;
total+=damt;
if(pr.rets_bal()<=(pr.rets_in()*10/100))
{
clrscr();
int dd;
char dnn[40];
cout<<"\nGetting low on stock, press ENTER to place an
order";
```

```
getch();
cout<<"\nProduct number: "<<pr.retpno();
cout<<"\nProduct name: "<<pr.retname();
cout<<"\nQuantity: "<<pr.rets_in()-pr.rets_bal();
cout<<"\nDealer No.: ";cin>>dd;
cout<<"\nDealer Name.: ";gets(dnn);
cout<<"\nM.R.P: "<<pr.retprice();
cout<<"\nDealer Price: "<<pr.retprice()*90/100;
cout<<"\nTotal:
"<<pr.retprice()*(pr.rets_in()-pr.rets_bal())*90/100;
pr.s_set();
}
}
fp.seekp((fp.tellg()-sizeof(pr)),ios::beg);
fp.write((char*)&pr,sizeof(product));
fp.read((char*)&pr,sizeof(product));
}
fp.close();
}
cout<<"\n\n\t\t\t\t\tTOTAL = "<<total;
getch();
}
//*****************************
//function to make stock report
//*****************************
void s_report()
{
clrscr();
fp.open("Shop.dat",ios::in);
if(!fp)
{
cout<<"ERROR!!! FILE COULD NOT BE OPEN\n\n\n
Go To Admin Menu to create File";
```

```
cout<<"\n\n\n Program is closing ....";
getch();
exit(0);
}
cout<<"\n\n\t\tStock Report\n\n";
cout<<"========================================
========================================\n";
cout<<"P.NO.\t\tNAME\t\tPRICE\tStock In\tStock
Out\tBalance\n";
cout<<"========================================
========================================\n";
while(fp.read((char*)&pr,sizeof(product)))
{
cout<<pr.retpno()<<"\t\t"<<pr.retname()<<"\t"<<pr.retpri
ce()<<"\t"<<pr.rets_in()<<"\t"<<pr.rets_out()<<"\t"<<pr.
rets_bal()<<endl;
}
fp.close();
}
//**************
// INTRODUCTION
//**************
void intro()
{
clrscr();
gotoxy(31,11);
cout<<"DEMRIS SUPER MARKET";
gotoxy(35,14);
cout<<"BILLING";
gotoxy(35,17);
cout<<"PROJECT";
cout<<"\n\nMADE BY : ABHISHEK SURESH";
```

```
cout<<"\n\nSCHOOL : MAHARISHI VIDYA MANDIR
SENIOR SECONDARY SCHOOL";
getch();
}
//*****************************
// ADMINSTRATOR MENU FUNCTION
//*****************************
void admin_menu()
{
clrscr();
char ch2;
cout<<"\n\n\n\tADMIN MENU";
cout<<"\n\n\t1.CREATE PRODUCT";
cout<<"\n\n\t2.DISPLAY ALL PRODUCTS";
cout<<"\n\n\t3.QUERY ";
cout<<"\n\n\t4.MODIFY PRODUCT";
cout<<"\n\n\t5.DELETE PRODUCT";
cout<<"\n\n\t6.VIEW PRODUCT MENU";
cout<<"\n\n\t7.BACK TO MAIN MENU";
cout<<"\n\n\tPlease Enter Your Choice (1-7) ";
cin>>ch2;
switch(ch2)
{
case '1': clrscr();
write_product();
break;
case '2': display_all();break;
case '3':
int num;
clrscr();
cout<<"\n\n\tPlease Enter The Product No. ";
cin>>num;
display_sp(num);
```

```
break;
case '4': modify_product();break;
case '5': delete_product();break;
case '6': menu();
getch();
case '7': break;
default:cout<<"\a";admin_menu();
}
}
//***************************
// DEALER ADMIN MENU FUNCTION
//***************************
void deal_menu()
{
clrscr();
char ch5;
cout<<"\n\n\n\tDEALER ADMIN MENU";
cout<<"\n\n\t1.ADD DEALER";
cout<<"\n\n\t2.DISPLAY ALL DEALER INFO";
cout<<"\n\n\t3.QUERY FOR DEALER";
cout<<"\n\n\t4.DELETE DEALER INFO";
cout<<"\n\n\t5.BACK TO MAIN MENU";
cout<<"\n\n\tPlease Enter Your Choice (1-5) ";
cin>>ch5;
switch(ch5)
{
case '1': clrscr();
write_dealer();
break;
case '2': display_all_d();break;
case '3':
int num1;
clrscr();
```

```
cout<<"\n\n\tPlease Enter The Dealer No. ";
cin>>num1;
display_sd(num1);
break;
case '4': delete_dealer();break;
case '5': break;
default:cout<<"\a";deal_menu();
}
}
//********
// Main()
//********
void main()
{
char ch;
intro();
do
{
clrscr();
cout<<"\n\n\n\tMAIN MENU";
cout<<"\n\n\t1. CUSTOMER";
cout<<"\n\n\t2. ADMINISTRATOR";
cout<<"\n\n\t3. STOCK REPORT";
cout<<"\n\n\t4. DEALER ADMIN";
cout<<"\n\n\t5. EXIT";
cout<<"\n\n\tPlease Select Your Option (1-3) ";
cin>>ch;
switch(ch)
{
case '1': clrscr();
place_order();
getch();
break;
```

```
case '2': admin_menu();
break;
case '3': s_report();getch();
break;
case '4': deal_menu();
break;
case '5':exit(1);
default :cout<<"Wrong choice"<<"\a";
}
}while(ch!='5');
}
// The end.
```

BIBLIOGRAPHY

1) Computer Science with C++ by Sumita Arora, Dhanpat Rai & Co.
2) Object Oriented Programming with C++ 3/e by E Balagurusamy, Tata McGraw Hill.
3) www.cplusplus.com