

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JnanaSangama, Belgaum -590014, Karnataka State, India



A MINI PROJECT REPORT ON “SUPERMARKET BILLING SYSTEM”

Submitted on partial fulfilment of academic requirement of 6th semester

File Structures with Mini Project Laboratory

BACHELOR OF ENGINEERING IN INFORMATION SCIENCE AND ENGINEERING

Submitted by:

CHANDAN R
(1SJ19IS021)

AKASH N
(1SJ19IS004)

CHANDAN KJ
(1SJ19IS020)

Under the guidance of:

Mrs. NANDINI S
Assistant Professor
Dept. Of ISE, SJCIT.

Carried out at:

**File Structures with Mini Project Laboratory,
SJCIT**



**S J C INSTITUTE OF TECHNOLOGY
Department of Information Science and Engineering
Chikkaballapur – 562101
2021-2022**

||Jai Sri Gurudev||

Sri AdichunchanagiriShikshana Trust®

S.J.C INSTITUTE OF TECHNOLOGY

Information Science & Engineering Department

Chikkaballapur-562101



CERTIFICATE

This is to certify that the Mini-project work entitled “**SUPERMARKET BILLING SYSTEM**” is a bonafide work carried out at File Structures with mini project laboratory by **CHANDAN R(1SJ19IS021)** **CHANDAN K J(1SJ19IS020)** and **AKASH N (1SJ19IS004)** in partial fulfilment for the **Bachelor of Engineering in Information science and Engineering in sixth semester of the Visvesvaraya Technological University, Belgaum** during the year **2021-22**. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect to sixth semester mini project work.

Signature of the guide

Mrs. Nandini S
Assistant Professor
Dept. of ISE, SJCIT

Signature of HOD

Prof. Sateesh Chandra Reddy
Head of the department
Dept. of ISE,SJCIT

Name of Examiner

Signature with Date

1)

2)

CONTENTS

Acknowledgement i

Abstract ii

Chapter No	Chapter Name	PAGE NO
------------	--------------	---------

CHAPTER 1	INTRODUCTION	1-3
------------------	---------------------	------------

1.1 Introduction to the project	1
---------------------------------	---

1.2 Reason for the project	1
----------------------------	---

1.3 Problem statement	2
-----------------------	---

1.4 Aims & Objectives	3
-----------------------	---

1.5 Scope of the object	3
-------------------------	---

CHAPTER	LITERATURE SURVEY	4-6
----------------	--------------------------	------------

2.1 Introduction to supermarket	4
---------------------------------	---

2.2Types of supermarket	4
-------------------------	---

2.3 Feature of existing system	5
--------------------------------	---

2.4 Description of new system	6
-------------------------------	---

CHAPTER3	SYSTEM REQUIREMENTS	7-8
-----------------	----------------------------	------------

3.1 Software Requirement	7
--------------------------	---

3.2 Hardware Requirements	7
---------------------------	---

3.3 Software features	8
-----------------------	---

CHAPTER4	DESIGN &ANALYSIS	9-14
-----------------	-----------------------------	-------------

4.1 System design	9
-------------------	---

4.2 Physical design	9
4.3 Data flow diagram	10
4.4 Interface design	11

CHAPTER 5	IMPLEMENTATION	12-24
------------------	-----------------------	--------------

5.1 System implementation	12
5.2 Changeover method	12
5.3 Code snippet	12

CHAPTER 6	SOFTWARE TESTING	25-29
------------------	-------------------------	--------------

6.1 Unit testing	25
6.2 System testing	25
6.3 White box testing	25
6.4 Black box testing	26
6.5 Snapshots	27

CHAPTER 7	CONCLUSION	30
------------------	-------------------	-----------

REFERENCES

ACKNOWLEDGEMENT

With Great pride we would like to convey my gratitude and appreciation to our alma-mater “**S.J.C Institute of Technology**” for giving us the required platform for the fulfillment of the Mini project on File structure as per the V.T.U requirements, for the fifth semester lab on the same subject.

We express my sincere thanks to **Dr.G.T. RAJU**, Principal of **SJCIT**, Chickballapur for providing me with excellent infrastructure to complete the Mini project.

We express whole hearted gratitude to **Prof. Sateesh Chandra Reddy** who is the respectable **HOD of Information Science and Engineering Department**. We wish to acknowledge his help in making my task easy by providing me with his valuable help and encouragement.

It is our pleasure to thank my guides **Mrs. Nandini S, Assistant Professor , Department of Information Science and Engineering**, SJCIT for their guidance, encouragement and valuable suggestion from the beginning of the Mini project work till the completion without which this Mini project work would not have been accomplished.

We are greatly indebted to them. And last but not the least, We would be very pleased to express my heart full thanks to the teaching and non-teaching staff of the Department of Information Science and Engineering, SJCIT for their motivation and support.

We also thank all those who extended their support and co-operation for successful completion of this Mini-Project.

CHANDAN R, CHANDAN KJ, AKASH N
1SJ17IS021, 1SJ19IS020, 1SJ19IS004

ABSTRACT

The project “supermarket billing system” deals with the automation of supermarket. This project helps to salesperson in managing the various type of records pertaining to the customer. The product will help the user to work in a highly effective environment.

The salesperson have recording the customer information in the past and even in present through their manual effort. And indeed ,it consumes their considerable time and energy that could be utilized in the better productive activities. Apart from that with increasing customer strength, the task of managing information of each individual customers indeed a cumbersome task.

There is a lot of reason for the introduction of this project. In this manual system ,there are number of inefficiencies that a salesperson faces. The information retrieval is one of the foremost problem. It is very difficult to gather the overall performance of the customer. Large book records have to be maintained where relevant and irrelevant information has to be stored which is very untidy and clumsy process.

On the other hand, there are many inherent that problem exist in any manual system. Usually they lack efficiency. Less efficiency has a great impact on the productivity of any human being keeping the up-to –date.

The project “supermarket billing system” is developed with the objective of the making the system reliable, easier , fast ,and more informative.

Chapter 1

INTRODUCTION

1.1 Introduction to project

Supermarket management system is the system where all the aspects related to the proper management of supermarket is done. These aspects involve managing information about the various products, staff, managers, customers, billing etc. This system provides an efficient way of managing the supermarket information. Also allows the customer to purchase and pay for the items purchased.

This project is based on the sales transaction and billing of items in a supermarket. The first activity is based on adding the items to the system along with the rate which are present in the supermarket and the name of the items which the supermarket will agree to sell. This authority is given only to admin (administrator). Any modifications to be done in the item name and the rate can be done only by admin. He also has the right to delete any item. As the customer buys the products and comes to the billing counter, the user is supposed to enter the item name he purchased and the quantity of the item he had purchased. This is not a huge a task.

This study is to produce software which manages the sales activity done in a supermarket, maintaining the stock details, maintaining the records of the sales done for a particular month/year. The users will consume less time in calculation and the sales activity will be completed within a fraction of seconds whereas manual system will make the user to write it down which is a long procedure and so paper work will be reduced and the user can spend more time on the monitoring the supermarket. The project will be user friendly and easy to use.

The system will display all the items whose name starts with the letter selected by the user. He can select out of those displayed. Finally a separate bill will be generated for each customer. This will be saved in the database. Any periodic records can be viewed at any time. If the stock is not available, the supermarket orders and buys from a prescribed vendor. The amount will be paid by deducting the total amount acquired in the sales activity.

1.2 Reason for the project

1. This system provides list of various products
2. There are various brands information along with the additional details
3. There is online application form where customer can choose their respective product.

4. There is one important functions provided where the information about the staff can be maintained.
5. There is database connectivity provided where each customer detail has been stored.
6. The system Provide functions of editing customer details.
7. Its Provide functions of editing product details.
8. Its Provide functions of editing staff details.

1.3 Problem statement

Being a Computer Technology student we had to go into the business department to learn some basic sales and supermarket management topics to increase our intellectual understanding on the project at hand it was really tasking.

Building a standard Supermarket management system was not an easy task looking at the problems of existing manual system.

The factors for these difficulties are:

1. Time Consumption: Manual systems are time consuming, as the business owner must keep track of Supermarket sales on a daily basis, while updating the system manually at the end of the day.
2. Poor Communication: A manual Supermarket system requires employees and managers to write down each time an item is removed from the Supermarket. If one employee forgets to mention that the last coffee product has been removed from the Supermarket, a manager expects the item to still be available for a customer during a sale. Compared with a technical Supermarket system, a manual Supermarket system does not help the communication in the workplace.
3. Physical Counts: A manual Supermarket system does not provide any number, as all numbers from the Supermarket are gained through physical Supermarket counts. One of the difficulties of running a manual Supermarket system is that physical Supermarket counts must be performed frequently to control the items in the Supermarket. This is time consuming and can cost the business money, if employees must come in to help out outside of business hours.
4. Daily Purchases: Keeping track of daily purchases is another difficult controlling measure with manual Supermarket systems. A manual Supermarket system requires the employees to write down the items sold during a single work day. This can be a difficult task, as one employee may lose the list of items sold or another may forget to write down a sale.
5. Ordering Supplies: A manual Supermarket system does not update at the end of the day with updated Supermarket

1.4 Aims & Objectives

Aim of study is

To design a Computerized Supermarket Management System to ascertain stock level of a supermarket, when to order for more goods, keep status and updates of transactions, thereby helping progress level, stock taking and managerial decisions,.

The objective are

1. To study the functions of Supermarket management system.
2. To explore the challenges being faced by the manual system.
3. To make a software fast in processing, with good user interface.
- 4.To ensure accurate statistics of product item.
5. For Easy record of goods in store and proper identification.

1.5 Scope of the object

This research work covers stock control, management and tends to correct anomalies in Supermarket business. It analyses opening of new stocks, stock updates and ability to view existing ones. It provides quick way of operation by capturing the manual process and automating them. This project is helpful to computerize the item transaction, sales activity record keeping which is a very huge task and maintaining the stock.

Chapter 2

LITERATURE SURVEY

2.1 Introduction to Supermarket

A supermarket is a large form of the traditional grocery store, it is a self-service shop offering a wide variety of food and household products, organized into aisles. It is larger in size and has a wider selection than a traditional grocery store but is smaller and more limited in the range of merchandise than a hypermarket or big-box market.

The concept of an inexpensive food market relying on large economies of scale was developed by Vincent Astor. He founded the Astor Market in 1915, investing \$750,000 of his fortune into a 165 by 125 corner of in the famous 95 Manhattan avenue, creating in effect, an open air mini-mall that sold meat, fruit, produce and flowers. The expectation was that customers would come from great distances ("miles around"), but in the end even attracting people from ten blocks away was difficult, and the market folded in 1917. The concept of a super market was developed by entrepreneur Clarence Saunders and his Piggly Wigglystores. His first store opened in 1916. Saunders was awarded a number of patents for the ideas he incorporated into his stores. The stores were a financial success and Saunders began to offer franchises. The Great Atlantic & Pacific Tea Company, which was established in 1859, was another successful early grocery store chain in Canada and the United States and became common in North American cities in the 1920s. The general trend in retail since then has been to stock shelves at night so that customers, the following day, can obtain their own goods and bring them to the front of the store to pay for them. Although there is a higher risk of shoplifting, the costs of appropriate security measures ideally will be outweighed by reduced labor cost.

2.2 Types of Supermarket

Supermarket is categorized into different type due to their size, scale, products offered, Store Format and Trends While people use the terms "Grocery Store"," Hypermarket" and "Biomarker" interchangeably to refer to retail food stores, industry watchers offer more specific guidelines about different types of Supermarket.

Grocery Store: A grocery store is a retail store that primarily sells food. A grocer is a bulk seller of food. Grocery stores often offer non-perishable food, with some also having fresh produce, butchers, delis, and bakeries. Large grocery stores that stock significant amounts of non-food products, such as clothing and household items, are called Supermarkets. Some large supermarkets also include a pharmacy and an electronics section, the latter selling DVDs, headphones, digital alarm clocks, and similar items.

Hypermarket: Is an advanced supermarket which has an additional department store. The result is an expansive retail facility carrying a wide range of products under one roof, including full groceries lines and general merchandise. In theory, hypermarkets allow customers to satisfy all their routine shopping needs in one trip. After the successes of supermarkets and hyper-markets and amid fears that smaller stores would be forced out of business, franchise laws that made it more difficult to build hypermarkets and also restricted the amount of economic leverage that hypermarket chains can impose upon their suppliers.

Bigboxmarket: Is a physically large retail establishment, usually part of a chain. The term sometimes also refers, by extension, to the company that operates the store. The store may sell general dry goods, it is generally inaccessible to pedestrians and often can only be reached by motor vehicles, the big-box store is regarded as unsustainable and a failure of urban planning.

2.3 Features of the Existing System

The current system operates manual supermarket management system, from stocks, products, ordering and purchases etc. recorded in a book. This is faced with errors, incompleteness, and insufficient data for analysis. Information regarding stocks, products, sales and purchases are still in black and white which is not properly organized and managed. From the wholesalers to retailer bills, tickets, vouchers, receipts of products are recorded in a book but further operations are not being properly handled. As a result it is difficult in processing, updating and managing.

The factors for these difficulties are:

Labor-Intensive: A manual Super Market management systems is that they can be highly labor-intensive to operate. They require continuous monitoring to ensure that each transaction is accounted for and that products are maintained at the appropriate stocking levels. It is also more difficult to share inventory information throughout the business, because the lack of computerization makes accessing inventory records a more cumbersome process. The time spent monitoring inventory levels could be used on more productive activities for the business.

Human Error: A manual Supermarket management system relies heavily on the actions of people, which increases the possibility of human error. People might forget to record a transaction or simply miscount the number of goods. This results in needless additional orders that increase the company's inventory carrying costs and use up precious storage space. Inaccurate physical counts could also result in not ordering enough of a product, meaning the business could run out of a crucial item at the wrong time.

Time Wasting: A manual Supermarket management system has a huge tendency of time wasting as the sales manager could have a lot to tackle while many customer seeks attention and this is really affecting the business.

2.4 Description of the New System

To reduce the shortcomings of the existing system there is a need to develop a new system that could upgrade the status of the current system which is manual and slow to the system that will be automatic and fast. The new system should be concerned with offering the requirements of the customer and the workers, the system should be reliable, easier, fast, and more informative.

The new system should possess the qualities stated below.

Qualities of the new System

1. Reduction in processing cost.
2. Error reduction.
3. Automatic posting.
4. Improve reporting.
5. Automatic production of the documents and Reports.
6. Faster response time.
7. Ability to meet user requirements.
8. Flexibility.
9. Reduced dependency.
10. Improves resource uses.
11. Reduction in use of the paper.
12. Reduction in Man Power.

The system is a desktop Windows application. The system will provide the following

Main features:

- Calculate the bill.
- Store how many products are sold.
- Store products and their prices and with other information.
- Change the Graphical User Interface of the system.

The System Can't

- Print out bills
- Manage promotion.

.Chapter 3

SYSTEM REQUIREMENTS

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

Characteristics of good SRS:

- Complete.
- Consistent.
- Feasible.
- Modifiable.
- Unambiguous.
- Testable.

3.1 Software Requirements:

Operating System	Windows 10
Languages	C++
Software	DEV C++
Compiler	GCC Compiler

3.2 Hardware Requirements:

Processor	Intel Core Processor
RAM	512MB(minimum)
Hard Disk	300MB(minimum)

3.3 Software Features:

- **C++**

C++ is a general-purpose programming language. It has imperative, object-oriented and generic programming features, while also providing facilities for low-level memory manipulation.

It was designed with a bias toward system programming and embedded, resource-constrained and large systems, with performance, efficiency and flexibility of use as its design highlights. C++ has also been found useful in many other contexts, with key strengths being software infrastructure and resource-constrained applications, including desktop applications, servers (e.g. e-commerce, web search or SQL servers), and performance-critical applications (e.g. telephone switches or space probes). C++ is a compiled language, with implementations of it available on many platforms. Many vendors provide C++ compilers, including the Free Software Foundation, Microsoft, Intel, and IBM.

- **SUBLIME TEXT**

Sublime Text is a proprietary cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and markup languages, and functions can be added by users with plugins, typically community-built and maintained under free-software licenses.

- **CODE BLOCKS**

Code::Blocks is a *freeC, C++ and FortranIDE* built to meet the most demanding needs of its users. It is designed to be very extensible and fully configurable.

Finally, an IDE with all the features *you* need, having a consistent look, feel and operation across platforms.

Chapter 4

DESIGN AND ANALYSIS

4.1 SYSTEM DESIGN

The system design is the not a step by step adherence of clear procedures and guidelines. Through, certain clear procedures and guideline have emerged in recent days, but still much of design work depends on knowledge .That is most creative and challenging phase of the system life cycle is system design. The term design describes a final system and the process by which it is developed. It refers to the technical specifications that will be applied in implementing the new system. It also includes the construction of programs and program testing.

In the design the designer faces the many problems. Many of these will be due to constrains imposed by the user or limitation of the hardware and software available in the market. Something, it is difficult to enumerates the complexity of the problem and solution thereof since the variety problem is so great and no solutions are exactly similar.

The first step is to determine how the output is to be produced and in what format.

Samples of the output (and input) are also presented. Second input data and master files (data base) have to be designed to meet the requirements of the proposed output. The operational (processing) phase are handled through program construction and testing, including a list of the programs needed to meet the system's objectives and complete documentation.

Finally, details related to justification of the system and an estimate of the impact of the candidate system on the user and the organization are documented and evaluated by management as a step toward implementation.

4.2 PHYSICAL DESIGN

The scope of the systems design is depend upon the framework for the new system developed. Clearly defined logical design for the new system which meets customer requirements has led to new techniques that attempt to do the following:

- Improve productivity of analysts and programmers.
- Improve documentation and subsequent maintenance and enhancements.
- Cut down drastically on cost overruns and delays.
- Improve communication among the user, analyst, designer, and programmer.
- Standardize the approach to analysis and design.

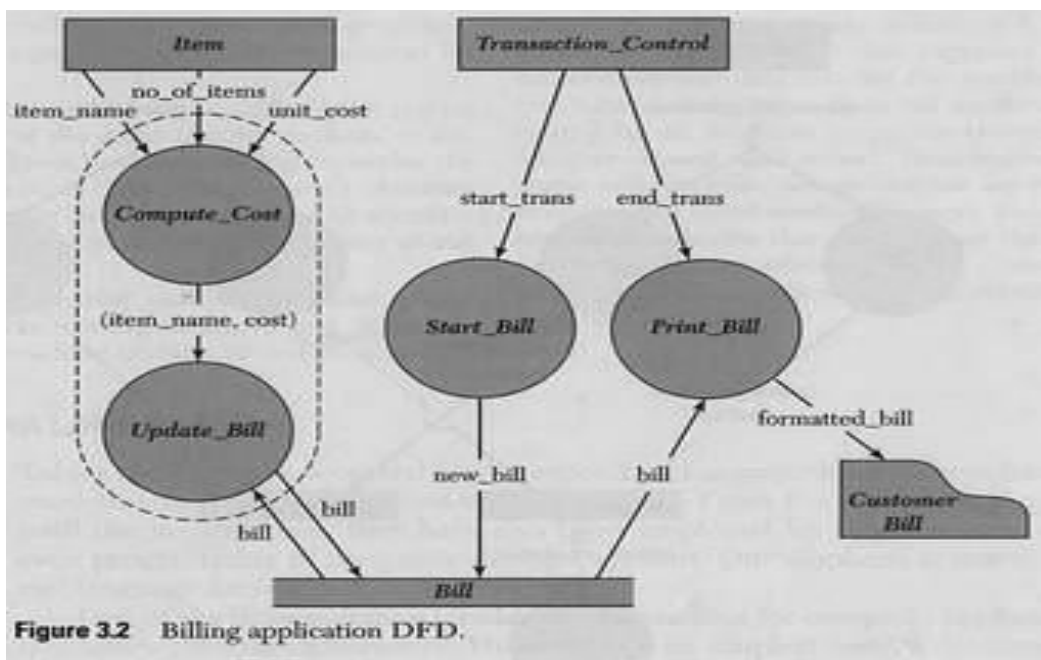
Simplify design by segmentation.

4.2.1 Processing Techniques

The processing options available to the designer are:

- Batch processing
- Real-time processing
- On-line processing
- A combination of all the above

4.3 Data flow diagram



4.4 INTERFACE DESIGN

4.4.1 OUTPUT DESIGN

Presenting the data processed by a system in a attractive and usable form has become very important. Therefore the system we must be know fully how to design output report in an attractive way.

There is there reason why the output form the system are required:-

- For communicating to the person concerned.
- processing.
- Foe permanent storage.

4.4.2 INPUT DESIGN

Once the output requirement have been finalized the next step is to find out what data need to be made available to the system to produce the desired outputs. If errors entered by data entry operates can be controlled by input design. Input design is the process of converting user-originated inputs to a computer based format. In the system design phase, the expanded data flow diagram identifies logical data flows, data stores, sources and destinations.

When the entering data operator needed to know the following:-

- The allocated space for each field.
- Field sequence , which must that in the document.
- The format in which data filed are entered .

Chapter 5

IMPLEMENTATION

Implementation is the realization of an application, or execution of a plan, idea, model, design, specification, standard, algorithm, or policy. In other words, an implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through programming and deployment. Many implementations may exist for a given specification or standard.

5.1 System Implementation

This chapter is the part that puts a planned system into action and examine in details the analysis and design of the Skillmid supermarket system. The present chapter discusses the implementation of the system, highlighting the testing exercise and describing some of the main components of the system's Graphical User Interface. It will give an output from programming language and other tools used to develop our system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system.

5.2 Changeover Method

This section deals with the strategy used to change from the old system to the new system. There are many methods available to swap from the old system to the new system these are direct changeover, parallel operation, pilot operation and phased operation.

I prefer a parallel operation changeover method for this system change over because this kind of method requires that both the old and the new information systems operate fully for a specified period. Data is input to both systems and output generated by the new system is compared with the equivalent output from the old system. When users, management, and IT group are satisfied that the new system (SKILLMID SUPERMAKET SYSTEM) operates correctly then the old system should be terminated. It is the most costly changeover method but it involves lower risks and it is very good.

5.3 CODE SNIPPET

```
#include<iostream>

#include<stdio.h>

#include<conio.h>

#include<process.h>
```

```
#include<fstream>

#include<string>

#include<stdlib.h>

using namespace std;

class product
{
    int prno;

    char name[50];

    float price,qty,dis,tax;
public:

    void createpr()
    {
        cout<<endl<<"Enter the product number"<<endl;

        cin>>prno;

        cout<<"Enter the product name"<<endl;

        std::cin>>name;

        cout<<"Enter the price of the product"<<endl;

        cin>>price;

        cout<<"Enter the discount percentage"<<endl;

        cin>>dis;
    }

    void showpr()
    {

        cout<<"Product number: "<<prno<<endl;

        cout<<"Product name: "<<std::cout<<name<<endl;

        cout<<"Product price: "<<price<<endl;
```

```
        cout<<"Discount percentage: "<<dis<<endl;
    }

    int retrprno()
    {
        return prno;
    }

    float retrprice()
    {
        return price;
    }

    char* retname()
    {
        return name;
    }

    int retdis()
    {
        return dis;
    }
};

fstream  fp;
product  pr;

int writepr()
{
    fp.open("SMBS.dat",ios::out|ios::app);
    pr.createpr();
    fp.write((char*)&pr,sizeof(product));
    fp.close();
}
```

```
    return 0;

}

int disp()
{
    cout<<"Display all the records: "<<endl;
    fp.open("SMBS.dat",ios::in|ios::out);
    while(fp.read((char*)&pr,sizeof(product)))
    {
        pr.showpr();
    }
    fp.close();
    return 0;
}

int dispsp()
{
    int flag=0,n;
    cout<<"Enter the product number: "<<endl;
    cin>>n;
    fp.open("SMBS.dat",ios::in|ios::out);
    while(fp.read((char*)&pr,sizeof(product)))
    {
        if(pr.retprno()==n)
        {
            pr.showpr();
            flag=1;
        }
    }
}
```

```
}

fp.close();

if(flag==0)

{

    cout<<"Record does not exist"<<endl;

}

return 0;

}

int modifypr()

{

    int no,found=0;

    system("cls");

    cout<<"Enter the product number:"<<endl;

    cin>>no;

    fp.open("SMBS.dat",ios::in|ios::out);

    while(fp.read((char*)&pr,sizeof(product))&&found==0)

    {

        if(pr.retprno()==no)

        {

            pr.showpr();

            cout<<"Enter the new details of the product:"<<endl;

            pr.createpr();

            int long long pos=-1*sizeof(pr);

            fp.seekp(pos,ios::cur);

            fp.write((char*)&pr,sizeof(product));

            cout<<"The details of the product has been modified"<<endl;

            found=1;

        }

    }

}
```

```
    }
}

fp.close();

if(found==0)

{
    cout<<"Record not found"<<endl;
}

return 0;
}

int deletepr()
{
    int num;

    system("cls");

    cout<<"Enter the product number of the product you want to delete"<<endl;
    cin>>num;

    fp.open("SMBS.dat",ios::in|ios::out);

    fstream fp2;

    fp2.open("Temporary.dat",ios::out);

    fp.seekg(0,ios::beg);

    while(fp.read((char*)&pr,sizeof(product)))
    {
        if(pr.retprno()!=num)
        {
            fp2.write((char*)&pr,sizeof(product));
        }
    }
}
```

```
fp2.close();

fp.close();

remove("SMBS.dat");

rename("Temporary.dat","SMBS.dat");

cout<<"The record of the product has been deleted"<<endl;

return 0;

}

int menu()

{

    system("cls");

    fp.open("SMBS.dat",ios::in);

    if(!fp)

    {

        cout<<"Error:The file cannot be opened, head over to the Admin menu to create a new file."<<endl;

        exit(0);

    }

    else

        cout<<"Product Menu:"<<endl;

    cout<<"....."<<endl;

    cout<<"Product number:\t\tName:\t\tPrice:"<<endl;

    cout<<"....."<<endl;

    while(fp.read((char*)&pr,sizeof(product)))

    {

        cout<<"\t"<<pr.retprno()<<"\t"<<pr.retname()<<"\t"<<pr.retprice()<<endl;

    }

    fp.close();
```



```
    return 0;
}

int placeorder()
{
    system("cls");

    int order[50],quan[50],c=0;

    float amt,damt,total=0;

    char ch='Y';

    menu();

    cout<<"\n\n\n\n..... ";

    cout<<"\n-Place your order-<<endl;

    cout<<"....."<<endl;

    do
    {
        cout<<"Enter the product number:<<endl;

        cin>>order[c];

        cout<<"Enter the quantity:<<endl;

        cin>>quan[c];

        c++;

        cout<<"Do want to order another product? (Y/N):<<endl;

        cin>>ch;

    }while(ch=='y'||ch=='Y');

    cout<<"Invoice:<<endl;

    cout<<"Product    number\tProduct    Name\tQuantity\tPrice\tAmount\tAmount    after
discount"<<endl;

    for(int x=0;x<=c;x++)
    {
```

```
fp.open("SMBS.dat",ios::in);

fp.read((char*)&pr,sizeof(product));

while(!fp.eof())

{

    if(pr.retprno()==order[x])

    {

        amt=pr.retprice()*quan[x];

        damt=amt-(amt*pr.retdis()/100);

cout<<"\n"<<"\t"<<order[x]<<"\t"<<pr.retname()<<"\t\t"<<quan[x]<<"\t\t"<<pr.retprice()<<
"\t"<<amt<<"\t\t"<<damt;

        total=total+damt;

    }

    fp.read((char*)&pr,sizeof(product));

}

fp.close();

}

cout<<"\n\n\t\t\t\tTotal:"<<total;

return 0;

}

int into()

{

    cout<<"-----Super Market Billing System Version 2 ----"<<endl;

    cout<<"\nMade by: code-projects.org"<<endl;

    return 0;

}

int admin()
```

```
{  
  
    system("cls");  
  
    char ch2;  
  
    cout<<"Administration menu:"<<endl;  
  
    cout<<"1.Create a product."<<endl;  
  
    //cout<<"2.Display all products."<<endl;  
  
    //cout<<"3.Query- Display a specific product."<<endl;  
  
    cout<<"2.Modify a product"<<endl;  
  
    cout<<"3.Delete a product."<<endl;  
  
    cout<<"4.View product menu."<<endl;  
  
    cout<<"5.Back to main menu."<<endl;  
  
    cout<<"Enter your choice:"<<endl;  
  
    ch2=getche();  
  
    switch(ch2)  
    {  
    case '1':  
        {  
            writepr();  
            break;  
        }  
    //case '2':  
        {  
            disp();  
            break;  
        }  
    //case '3':  
        {  

```

```
        system("cls");

        dispsp();

        break;

    }

case '2':

    {

        modifypr();

        break;

    }

case '3':

    {

        deletepr();

        break;

    }

case '4':

    {

        menu();

        getch();

    }

case '5':

    {

        break;

    }

    default: cout<<"\a";admin();

}

return 0;

}
```

```
int main()
{
    char ch;

    into();

    do
    {
        cout<<"\n\nWelcome to The Super Market Billing System Application"<<endl;

        cout<<"Main Menu:"<<endl;

        cout<<"1.Customer"<<endl;

        cout<<"2.Administrator"<<endl;

        cout<<"3.Exit"<<endl;

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

        ch=getche();

        switch(ch)
        {
            case '1':
                {
                    placeorder();

                    getch();

                    break;
                }

            case '2':
                {
                    admin();

                    break;
                }

            case '3':
```

```
{  
    exit(0);  
}  
default:cout<<"\a";  
}  
system("cls");  
}while(ch!=3);  
return 0;  
}
```

Chapter 6

SOFTWARE TESTING

6.1 UNIT TESTING:

In the unit testing the analyst tests the program making up a system. For this reason, unit testing is sometime called program testing. Unit testing gives stress on the modules independently of one another, to find errors. This helps the tester in the detecting errors in coding and logic that are contained within the module alone. The errors resulting from the interaction between are initially avoided. For each, it provides the ability to enter, modify retrieve data and respond to different type of inquiry or print report.

Unit testing can be performed from the bottom up, starting with smallest and lowest level modules and proceeding one at a time. For each module in bottom-up testing a short program is used to execute the module and provide the needed data, so that the module is asked to perform the way it will when embedded within the larger system.

6.2 SYSTEM TESTING:

The important and essential part of the system development phase, after designing and developing the software is system testing. We cannot say that every program or system design is perfect and because of lack of communication between the user and the designer, some error is there in the software. The number and the nature of error in a newly design system depend on some usual factors like communication between the user and the design.

Testing is an important function to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully activated. Another reason for system testing is its utility as a user-oriented vehicle before implementation.

6.3 WHITE BOX TESTING:

White-box testing or Glass-box testing is a test case design that uses control structure of the procedural design to derive test cases. The use of control structure was made to derive test cases for our modules also.

Following should guarantee that all the independent paths within a module should be exercised at least once.

All logical decisions should be exercised at to TRUE and FALSE sides.

All loops are traced by at least one test case to their boundaries.

All internal data structures are exercised for their validity.

Thereby, Conditional testing and Data flow testing have been exercised that are described as follows:

- **Conditional testing**

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generated on particular condition is traced to uncover any possible errors.

- **Loop Testing**

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

All the loops were tested at their limits, just above them and just below them.

All the loops were skipped at least once.

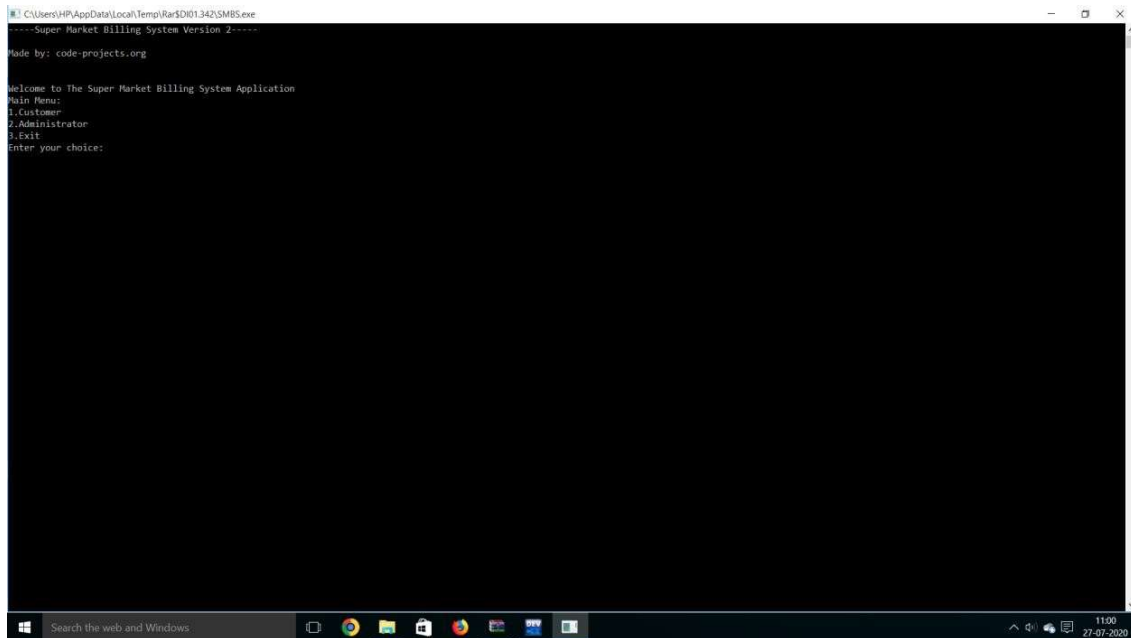
6.4 BLACK BOX TESTING:

Black box testing or behavioral testing is a testing method that tests the functional requirements of the Software. Though not a substitute for White-box testing but Black box testing is a good complimentary of White-box testing and concentrates on points with higher probability of errors.

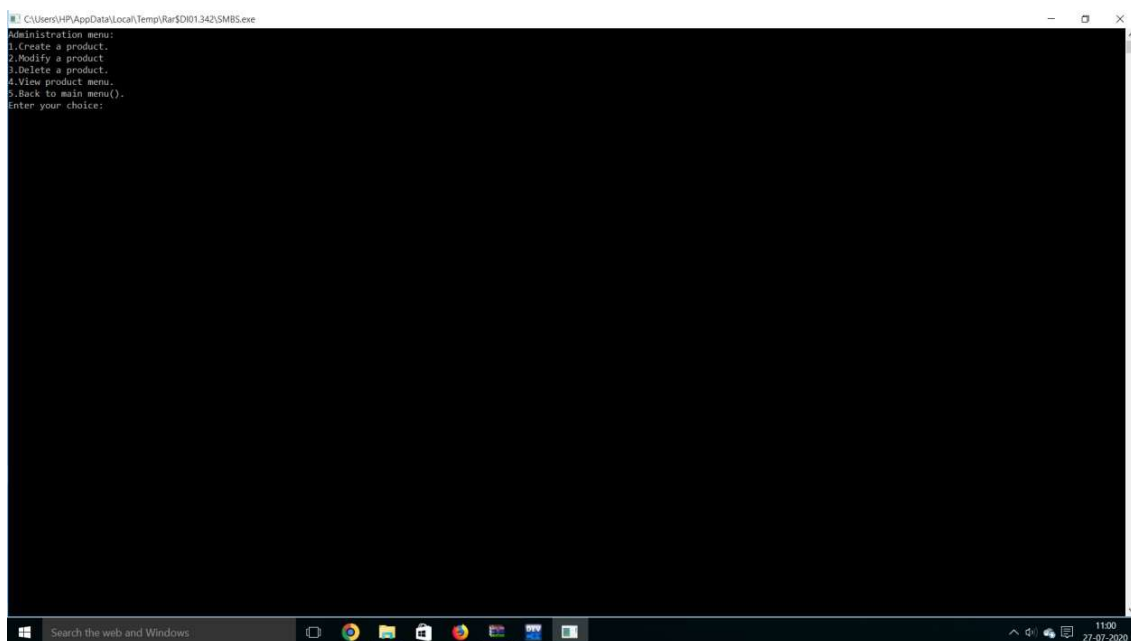
In this exercise, following are aimed specifically:

- Find Incorrect or Missing Functions
- Performance Errors
- Initialization and Termination Errors

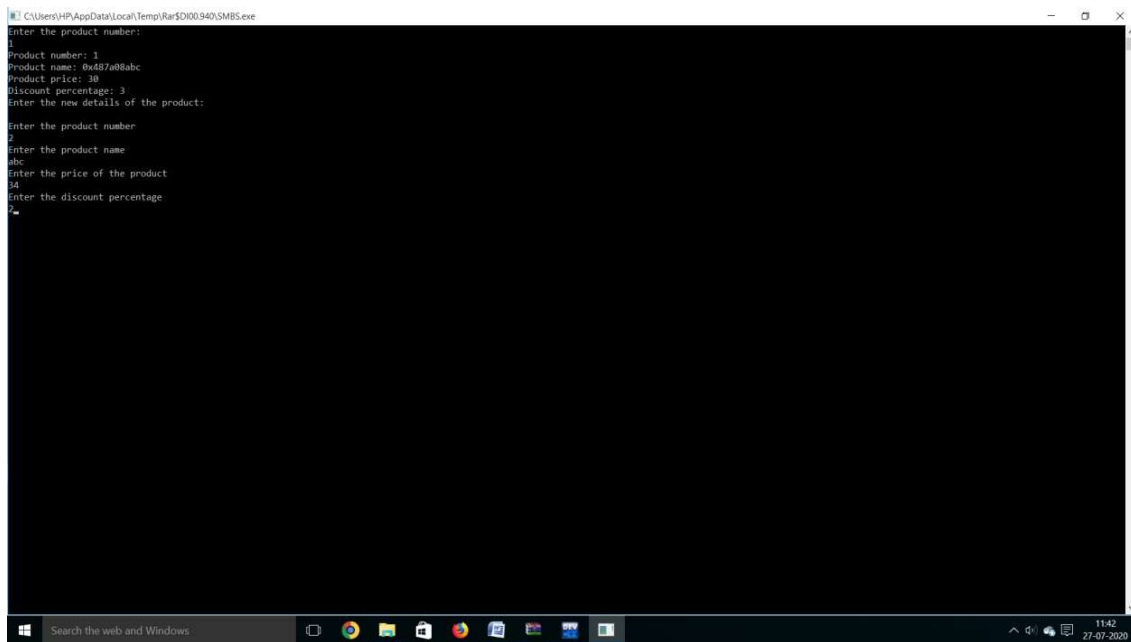
6.5 SNAPSHOTS



Menu page

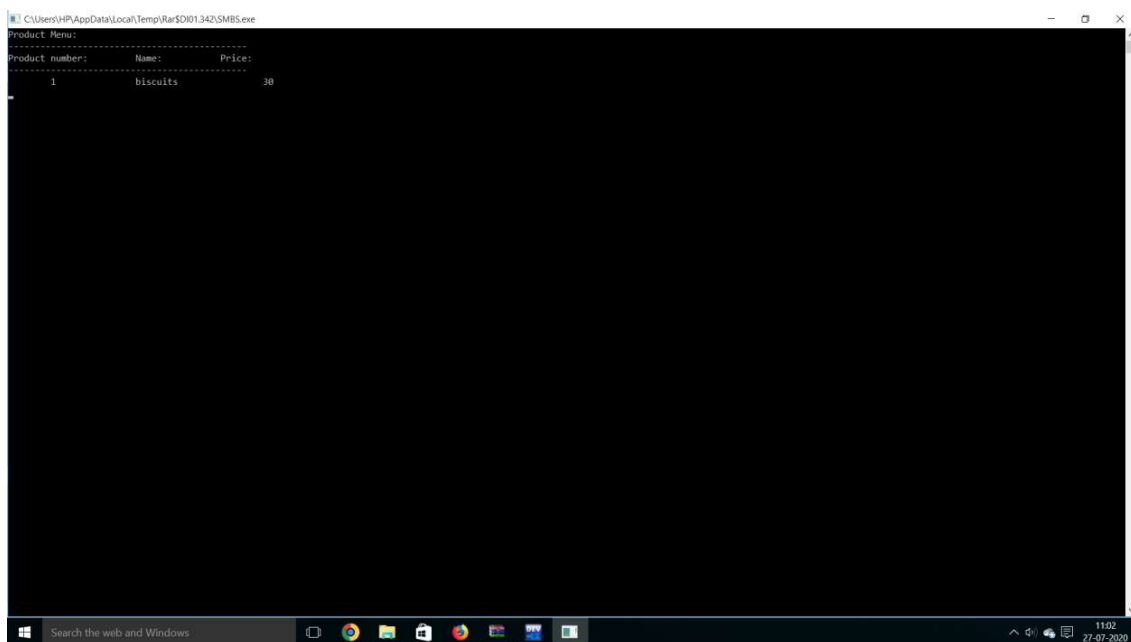


Administration menu page



```
C:\Users\HP\AppData\Local\Temp\Rar$DI00.940\SMBS.exe
Enter the product number:
1
Product number: 1
Product name: 0x487a088abc
Product price: 30
Discount percentage: 1
Enter the new details of the product:
Enter the product number
2
Enter the product name
abc
Enter the price of the product
54
Enter the discount percentage
2
```

Modify the item



```
C:\Users\HP\AppData\Local\Temp\Rar$DI01.342\SMBS.exe
Product Menu:
-----
Product number:   Name:   Price:
-----
1                biscuits 30
-----
```

Product menu page

```
C:\Users\HP\AppData\Local\Temp\Rar$001342\SMBS.exe
Product Menu:
-----
Product numbers:   Name:      Price:
-----
      2           biscuits    30

-----
-Place your order-
-----
Enter the product number:
2
Enter the quantity:
3
Do want to order another product? (Y/N):
n
Invoice:
Product number  Product Name  Quantity  Price  Amount  Amount after discount
-----
      2           biscuits      3         30         90         88.2
                                     Total:88.2

-----
Search the web and Windows
```

Placing the order

```
C:\Users\HP\AppData\Local\Temp\Rar$001342\SMBS.exe
Welcome to The Super Market Billing System Application
Main Menu:
1.Customer
2.Administrator
3.Exit
Enter your choice:
3
-----
Process exited after 841 seconds with return value 0
Press any key to continue . . .

-----
Search the web and Windows
```

Exit page

Chapter 7

CONCLUSION

In conclusion, Supermarket Management System has to do with making appropriate effort to stop the rising problem to all manual supermarket operation in order to enhance the operation of such supermarket. In this project, the software or system that can be used to aid all supermarkets that is still operating manually have been successfully developed. The software can be implementing in all types of supermarket as mentioned in the second chapter. The software has a large memory of storing all the goods in the supermarket and also keeping record it is highly effective and accurate.

In the development of this supermarket management system, I will recommend that if there is going to be any modification the new writer should endeavor to improve on the limitations such as inclusion of the billing and printing to further increase the system architecture and to satisfy users need more for writing of the source code, visual studio 2012 should be used and Microsoft access for the database. There are some limitations during the development of this supermarket management system that will require improvement as stated in previous chapter writer should put them in mind and face it as a challenge and not a problem.

In the future, the following components can be added to the system in order to improve the effectiveness and efficiency of the system, which includes:

1. An advanced password system that will be embedded into all login pages to increase the security of the system.
2. A good Printing module should be included.
3. A good internet backup should be automated after everyday sales.
4. Internet Transactions should be allowed.

A lot of challenges surfaced during the development of this incredible application though it tried stopping this project but the doggedness and consistency of the writer was in match with the challenge .

REFERENCES

- 1) The C++ Programming Language by Bjarne Stroustrup.
- 2) File Structures-An object oriented approach with C++. Pearson Education 3rd Edition.
- 3) File Structures : An Object-Oriented Approach with C++ 3rd Edition]
- 4) www.wikipedia.com
- 5) www.geeksforgeeks.com

