Ministry of high education,
Culture and science city at Oct 6,
The High institute of computer Science & information systems

المعهد العالي لعلوم الحاسب و نظم المعلومات

Graduation Project:

Super Market Management System

(SMMS)

Supervised by:

Prof. Dr. Ayman El Sayed

Assistant:

Eng. Ahmed Gaber

Project no. 22 **Academic Year: 2021 / 2020**

Ministry of high education,

Culture and science city at Oct 6,

The High institute of computer Science & information systems

المعهد العالي لعلوم الحاسب و نظم المعلومات

Graduation Project:

Super Market Management System (SMMS)

Prepared by

52003	علي محمد حمد عبدالله علي	51020	احمد ناصر سيف الدين حسين سليمان
52005	مؤمن محمد سعيد محمد محمد	51021	أحمد يسري عباس عبد المجيد
52009	محمد محمود عبدالله الجبالي	51028	إسماعيل أنور عبد الموجود أبوطالب
52010	محمود عصام فهمي عبد الغني	51076	محمد عادل هندي أمين

Supervised by:

Prof. Dr. Ayman El Sayed

Assistant:

Eng. Ahmed Gaber

Project no. 22 **Academic Year: 2021 / 2020**

Acknowledgement

In performing our project, we had to take the help and guideline of some respected persons, who deserve our greatest gratitude. The completion of this assignment gives us much pleasure. We would like to show our gratitude to **Prof.Dr. Ayman El Sayed** for giving us a good guideline for assignment throughout numerous consultations. We would also like to expand our deepest gratitude to all those who have directly and indirectly guided us in writing this assignment.

In addition, a thank you to **Eng. Ahmed Gaber** introduced us to the methodology of work.

Many people, especially our classmates and team members themselves, have made valuable comment suggestions on this proposal which inspired us to improve our assignment. We thank all the people for their help directly and indirectly to complete our assignment.

Abstract

The proposed supermarket management is windows based application system. The main objective of this application is to make it interactive and its ease of use for medium and mini supermarkets by making the system reliable, fast, user-friendly, and informative. It reduces paperwork, manpower requirement, and increases the productivity of the supermarket.

Using this application, one can add, modify, update, save, delete, and print details. There's also a search feature to find products available in the supermarket.

Table of contents

List of figures

List of tables

Chapter

1

INTRODUCTION

Overview

The increasingly critical nature of projects such as supermarket management system makes them resort to project management software solutions. It has also been noted that the inability to handle the increasing complexity of the project without the help of a software solution increases the possibility of project failure.

Figure (1) - Supermarket

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 task.

This study is to produce a 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.

1.2 The Modules of Operation is stated below

Account Configuration:

1. **Employee:** When a new employee joins the company, his record is saved in the database.
2. **Items:** Here the Admin can add any new items present in the supermarket. He also has the right to modify or delete it from the database.
3. **Registration:** As soon as the employee joins the company, the admin provides unique username and password to him.
4. **Vendor Order:** 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.
5. **Stock entry:** The items bought from the vendor will be entered here and this will be added to the stock.
6. **Indent Report:** This provides the report of the items sold for a particular month/ year and also gives the total amount acquired.
7. **Vendor Report:** This provides the report of the items bought from a vendor for a particular month/ year and also gives the total amount spent.
8. **Display:** A user can view information regarding Items present in the supermarket.

9. **Logout:** This module allows the user to Logout the application. Further operations cannot be performed after user exits.

1.3 Function of the System

1. This system provides list of various products
2. There are various brands information along with the additional details
3. There is one important functions provided where the information about the staff can be maintained.
4. There is database connectivity provided where each customer detail has been stored.
5. The system Provide functions of editing customer details.
6. Its Provide functions of editing product details.
7. Its Provide functions of editing staff details.

1.3 Project objective

The objective of this application is to develop a Windows-based application for supermarket management. The system will be easy to use and thus make the management experience enjoyable for the users. This application aims to develop an easy-to-use Windows-based interface that advises users to check the stock level in the supermarket when to order more goods, keep status and transaction updates, thus assisting with progress, inventory and management decisions, searching for products, and viewing descriptions The entirety of sales, purchases, costs or profits.

The objectives are to;

1. Study the functions of Supermarket management system.
2. Explore the challenges being faced by the manual system.

3. Make a software fast in processing, with good user interface.
4. Ensure accurate number of product item that are available.
5. Have easy record of goods in store and proper identification.

1.4 Project scopes

The project **scopes** 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.

1.5 Limitation of project

Due to time and basic factors like unstable electricity, poor networks, unavailability of concrete business idea and many more this research has been limited to certain areas in supermarket management we only looked more into the supermarket inventory management area using MYSQL, Visual C# and Visual Studio.

1.6 Definition of Terms

i. **MODULES:** This can be described a search of a set of standardized parts or independent units that can be used to construct a more complex structure, such as an item.
ii. **ADMIN:** This is the administration of a business, organization,
iii. **USERNAME:** This is an identification used by a person with access to a computer, network, or online service.

iv. **PASSWORD:** This is a secret word or phrase that must be used to gain admission to a place.

v. **VENDOR ORDER:** This is a commercial document used to request someone to supply something in return for payment and providing specifications and quantities;

vi. **INDENT REPORT:** This contains Order for goods (placed often through a local or foreign agent of a foreign supplier) under specified conditions of sale, the acceptance of which by the supplier (or the agent) constitutes a contract of sale.

vii. **Microsoft VISUAL STUDIO:** Visual Studio (VS) is an integrated development environment (IDE) is a set of tools in a single application that helps you write programs.

viii. **Microsoft C#:** Microsoft C# (pronounced C sharp) is a new programming language designed for building a wide range of enterprise applications that run on the .NET Framework

ix. **MYSQL:** MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use.

1.7 Platform Specifications – Deployment

1.7.1 Hardware Specification

- Processor Pentium VI
- RAM 8 GB
- Minimum Space Required 10 GB
- Display 16 bit color

1.7.2 Software Specifications

- Operating Environment Win 2000/8
- Platform .Net Framework & IIS Visual Studio 2015
- Database SQL Server 2018

Chapter

Theoretical Background and tools

2.1 Challenges 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:

i. **Labor-Intensive:** A manual supermarket 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.

ii. **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.

iii. **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.

2.2 Description/Analysis 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 concern 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.

2.2.1 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:

1. Calculate the bill.

2. Store how many products are sold.
3. Store products and their prices and with other information.
4. Change the Graphical User Interface of the system.

The System Can't

1. Print out bills
2. Manage promotion
3. Compute more than one item at the same time

Due to the following reason: This project is based on the sales transaction and Stocking of items in a supermarket.

2.3 Methods of Data Collection

For the system to meet management, staff and customer needs once implemented, the team of experts conducted a thorough data collection process using this method:

Observation Method.

This was closely observed that supermarket activities as the customers are served at the supermarket and how the management carried out its managerial activities. It was found out that the supermarket indeed was in a need for automation of its activities to realize its maximum potential. It was also observed that when a customer buy a product all the details was not well managed by the current manual system, we came up with this system to clearly address these problems while managing the supermarket.

2.4 Architectural Design of the Proposed System

The system will cover two part which are the Administrative part and the Users part

2.4.1 Administrative part: Add staff, Registered Staff, Add Item, Search Item, Report, Warehouse and Logout.

1. Add staff

   Contains a form that the administrator of the system can add a new user of the supermarket management system as a staff.
2. Registered Staff

   Contains the details of all staff (users) of the system
3. Add Item

   Contains the form for adding a new item that have been purchase by the supermarket and kept it in the store of the system.
4. Search Item

   Responsible for checking of an item from the store to see if the item is available in the supermarket
5. Report

   Shows all the thing that were purchase in the supermarket. This includes general reports, such as search by name, by date and print.
6. Warehouse

   Contains a large store of data accumulated from a wide range of sources within a company and used to guide management decisions.
7. Logout

   It will take the administrator out of the home page of the Admin page in the system

2.4.2 User's part: Product, Repot and Logout.

1. Product

   This is where the staff (user) use to put the details of the purchase item in the supermarket
2. Report

   Shows all the thing that were purchase in the supermarket. This includes general reports, such as search by name, by date and print.
3. Logout

   It will take the staff (user) out of the home page of the user's page in the system.
4. Exit Program. It will close the whole program.

System Design is one of the tasking sections of the Programming. In this section of the project many previews are going to be seen and we are gradually getting close to the new system. System design is a transition from a user-oriented document to a document oriented to programmers or database personnel. The system design is structured into the following parts:

i. Input design
ii. Output design
iii. Database design
iv. System Flowchart

2.4.3 Input Design

In any organization, institution or any system of operation there is always an input into the system which keeps a system going, if the input is wrong definitely the output will be wrong. This design is meant to handle data about a particular product or stock in the Supermarkets as shown in figure 3.1

Customers Form

Surname

Other name

Gender

Occupation

Contact Address

Phone No.

Staff_id

Item_type

Item_name

Item_no

Item_size

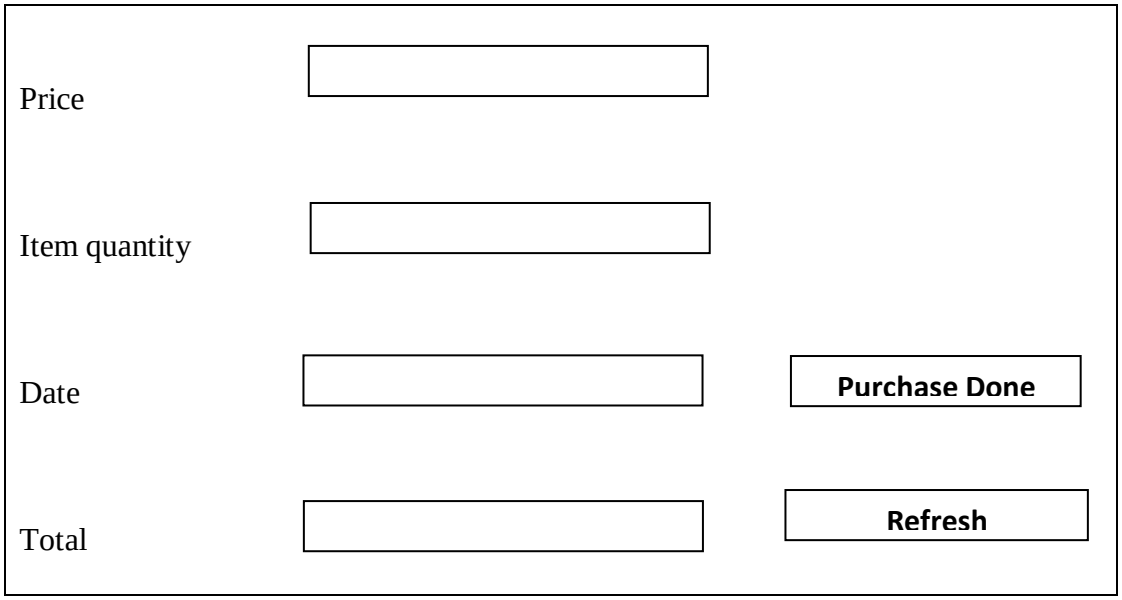


Figure 2.1

2.4.4 Output Design

In a very competitive world that we are, a good and attractive GUI is needed to make customers and administrators enjoy the services of a system, which would serve as a system to increase productivity in supermarket business.

The system will use reports to output the information as shown in Table 2.1

Report

Surname	Other_name	Gender	Occupation	Contact_Address	Phone_no
Emmanuel	Bitrus	Male	Student	Ecwa Goodnews	07064749825
Yusuf	Ishaku	Male	Business	Billiri	Normal

Staff_id	Item_type	Item_name	Item_no	Item_size	Price
Sid-03	Provision	Maggie	101	Small package	320
Sid-05	Accessories	Gionee P2	403	5inch	14000

Item_quantity	Date	Total
12	22/09/2016	N38400
4	27/08/2016	56000

Table 2.1: Table for the Output Design for customers.

2.4.5 Database Design

Database is a file composed of records, each containing fields together with a set of operations it helps in organizing data in a logical order for references.

Database contains related data which are organized together in a group of object, table, and file. It can be in form of node.

The Database design of this system is showed in Table 3.2 – 3.4 while the system flowchart is shown in the diagram 3.1

Field Name	Data Type	Field Size
Id	Int	11
Staff Id	Varchar	8
First_name	Text	15
Surname	Text	15
Last_name	Text	15
Gender	Text	8
Date_of_birth	Varchar	15
Department	Text	20
Nationality	Text	15
State	Text	10
Lga	Text	10

Address	Varchar	100
Email	Varchar	50
Phone No.	Int	15
Year_of_appointment	Varchar	12
Qualification	Text	15
Password	Varchar	15
Passport	Mediumblob	
Kin_first_name	Text	15
Kin_surname	Text	15
Kin_last_name	Text	15
Kin_nationality	Text	15
Kin_state	Text	10
Kin_lga	Text	10
Kin_address	Varchar	100
Kin_relationship	Text	15
Kin_email	Varchar	50
Kin_phone_no	Int	15

Table 3.2: Table for the Database Design

Field	Data type	Field Size
Id	Int	11
Username	Varchar	12
Password	Varchar	12

Table 3.3: Table for the Databaset Design to login

Field Name	Data type	Field Size
Id	Int	11
Item_type	Varchar	30
Item_no	Varchar	10
Item_name	Varchar	15
Item_size	Text	15
Item_quantity	Int	5
Date	Varchar	15
Item_amount	Varchar	13

Table 3.4: Table for the Database Design to Add items to the Stock

2.5 System Flowchart

This is the logical structure that represents the blue print of proposed system in other words, it defines as the algorithm of the software in a concise and logical order. The process design is represented diagrammatically in the form of system flow chart as shown below

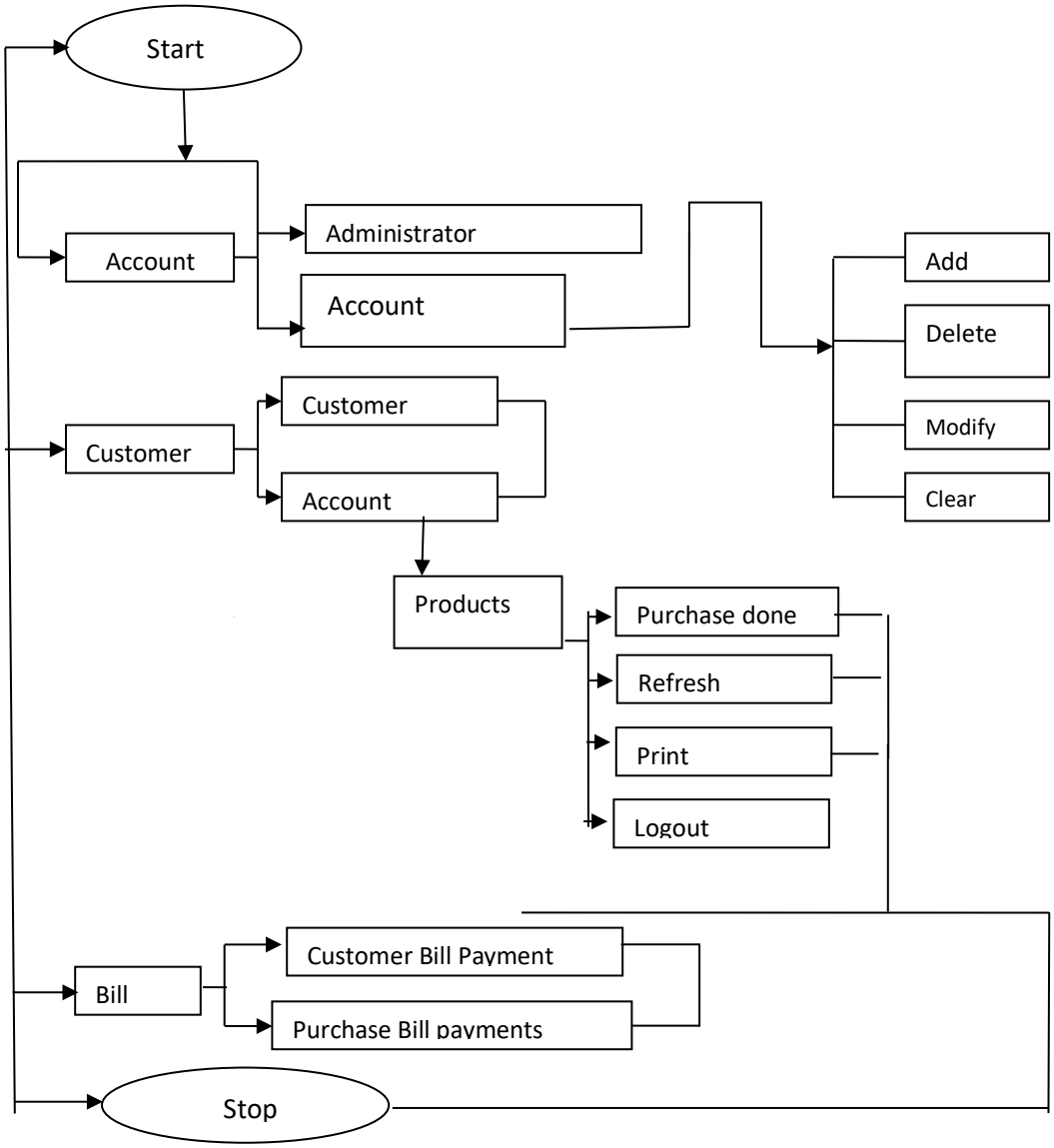


Figure 3.2 System Flowchart

2.6 Entity Relationship Diagram

An ER diagram is a pictorial representation of the information that can be captured by a database. Such a "picture" serves two purposes:

A. It allows database professionals to describe an overall design concisely yet accurately.

B. (Most of) it can be easily transformed into the relational schema.

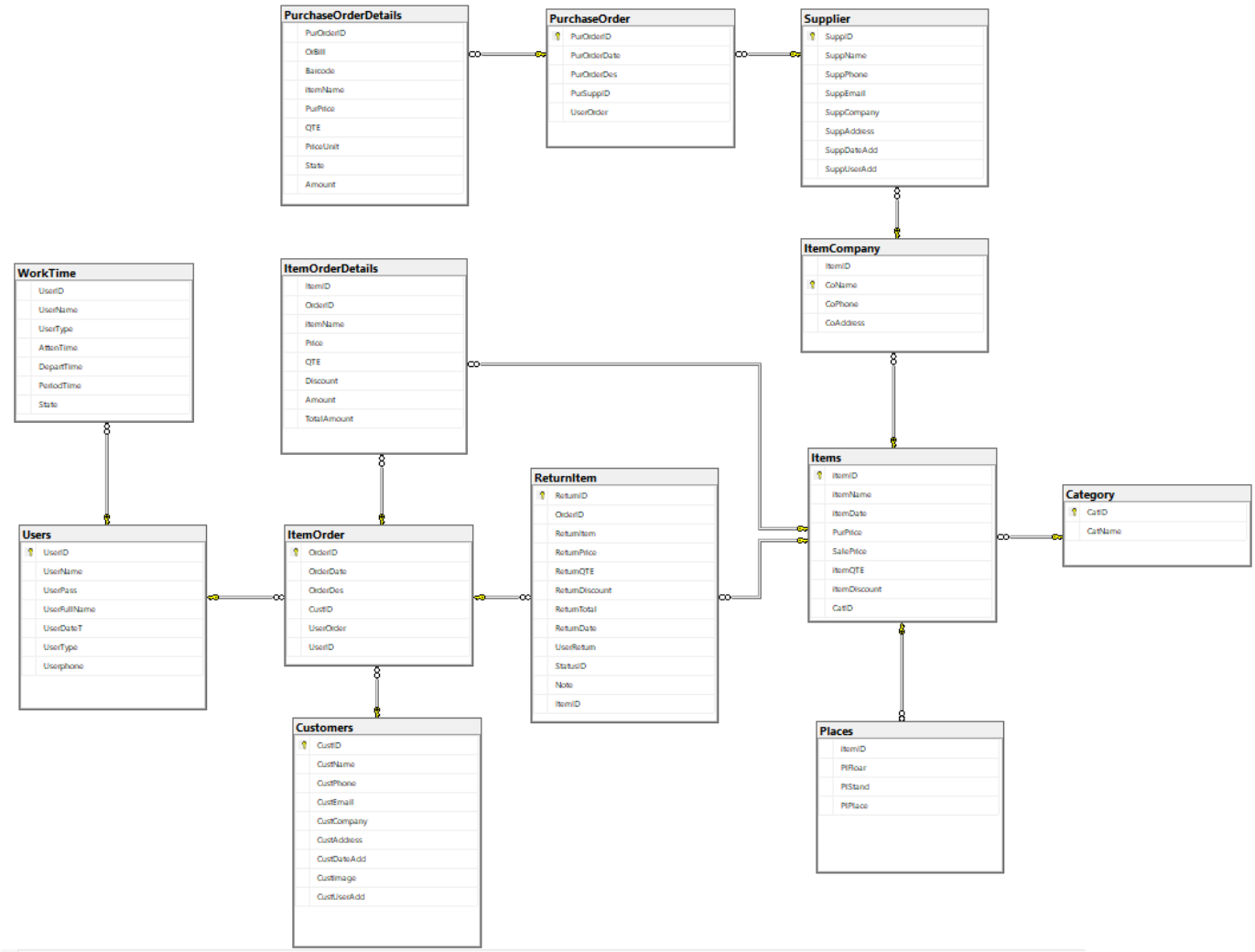


Figure 3.3 System ER Diagram

2.7 Use Case Diagram

They are usually referred to as behavior **diagrams used** to describe a set of actions (**use** cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). A ***use case diagram*** is a graphic description of the interactions among the elements of a system. A ***use case*** is a methodology ***used*** in system analysis to identify, clarify, and organize system requirements.

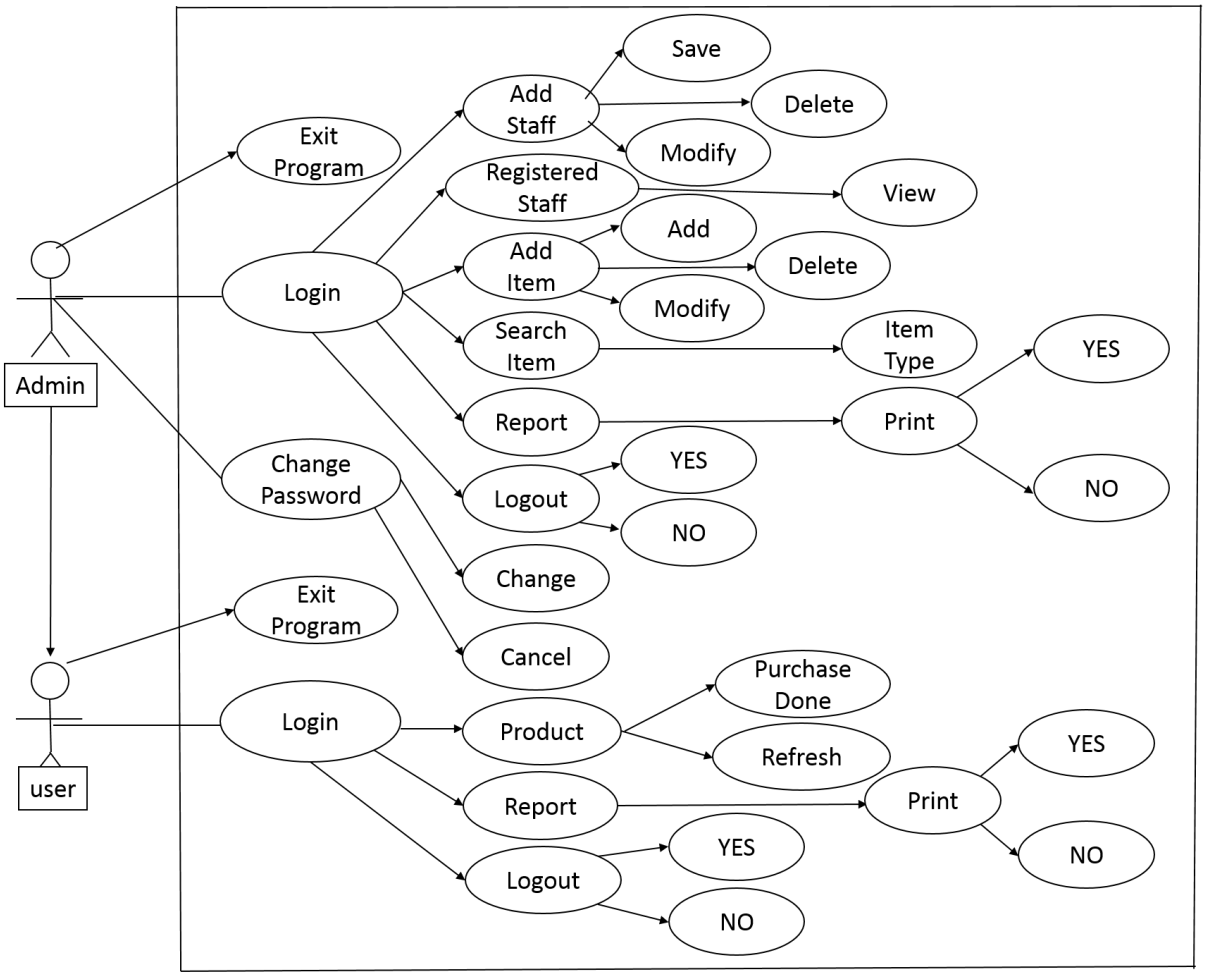


Figure 3.4 System Use Case

2.8 Materials

This has to do with the requirements for the effective functioning of the system this requirement can be viewed from a hardware and software perspective, the hardware requirement encompass the physical component that are required and the software requirement has to do with all that is required from a software level.

2.8.1 Hardware Specification

For a system to be used efficiently and accurately, all computer software needs certain hardware components or other software resources to be present on a computer. These prerequisites are known as (computer hardware specification) and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements. A second meaning of the term of System requirements is a generalization of this first definition, giving the requirements to be met in the design of a system or sub-system. Typically an organization starts with a set of Business requirements and then derives the System requirements from there. The most common set of requirements defined by any operating system or software application is the physical computer resources. This is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements for this application software

The hardware required include:

i. Processor 2.4 GHZ processor speed
ii. Disk space 80 GB (including 20 GB for database Management system)
iii. SVGA color monitor or higher quality.
iv. RAM 512MB.
v. Backup storage hard disk of about 80MB.
vi. Flash drive for file transfer.
vii. An enhanced keyboard.
viii. A power stabilizer.

2.8.2 Software Specification

This are requirements specification for a software system, is a description of the behavior of a system to be developed and may include a set of use cases that describe interactions the users will have with the software. In addition it also contains non-functional requirements.

Non-functional requirements impose constraints on the design or implementation such as performance engineering requirements, quality standards

Software requirements specification establishes the basis for agreement between customers and contractors or suppliers (in market-driven projects, these roles may be played by the marketing and development divisions) on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.

The software requirements specification document enlists enough and necessary requirements that are required for the project development. To derive the requirements we need to have clear and thorough understanding of the products to

be developed or being developed. This is achieved and refined with detailed and continuous communications with the project team and customer till the completion of the software.

The software components used for this project are:

i. Operating system; **Windows** 7/8/**10**.
ii. **Microsoft Visual Studio 2013** (Front end)
iii. **MySQL** Database (Back end)
iv. **Visual C#** Programming

Chapter

Project Implementation

3.1 Introduction

Result is something that follows naturally from a particular action, operation or course a consequences or outcome. Result is actually a output of a particular operation in this context. This Supermarket Management System will performs many task such as authentication of user and admin of the supermarket, adding new staff to the supermarket if a new staff is employed, sales of product to the customers and also the report of the sales and available product that are in the supermarket will all be seen.

3.2 Main form

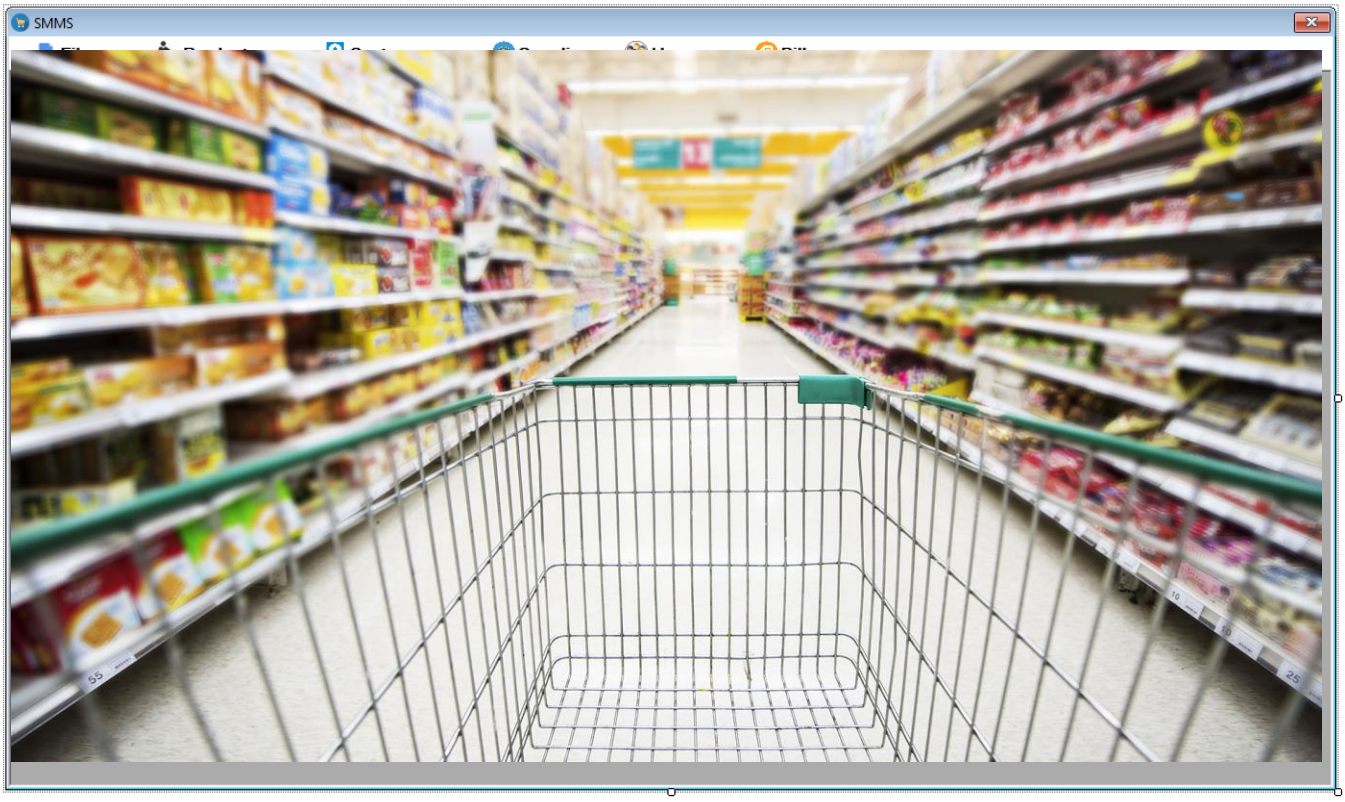


Figure 4.1: Start Page

3.3 Sign In – Sign Out form

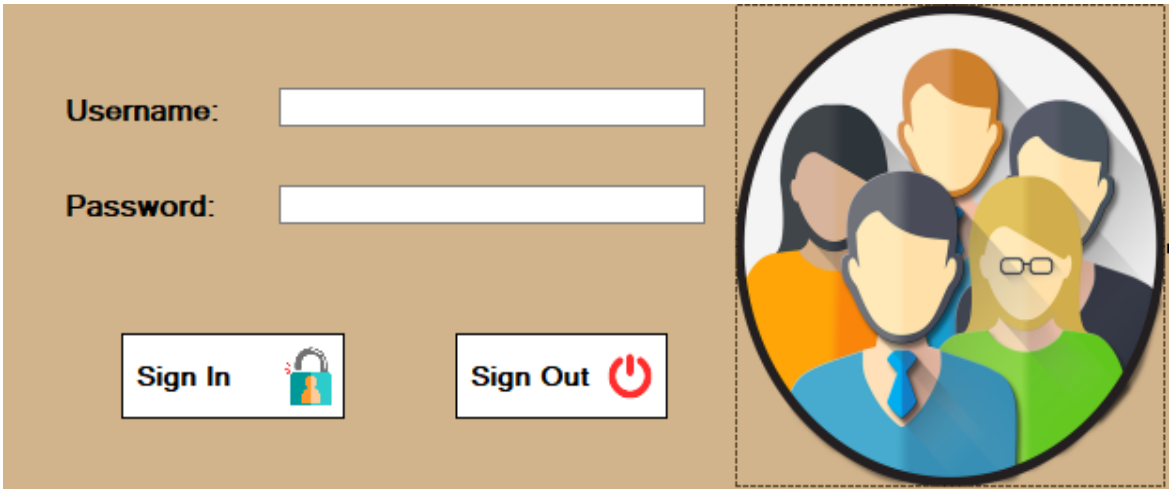


Figure 4.2: Login Page

3.4 sale operation form

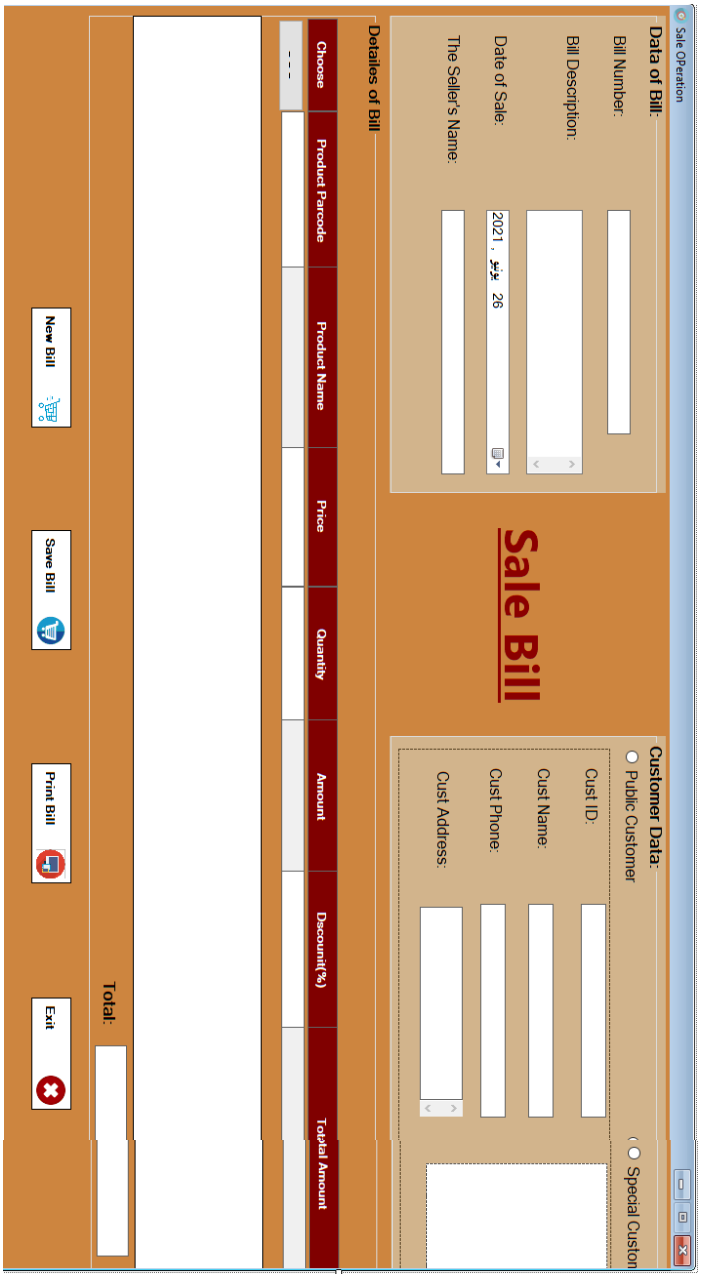
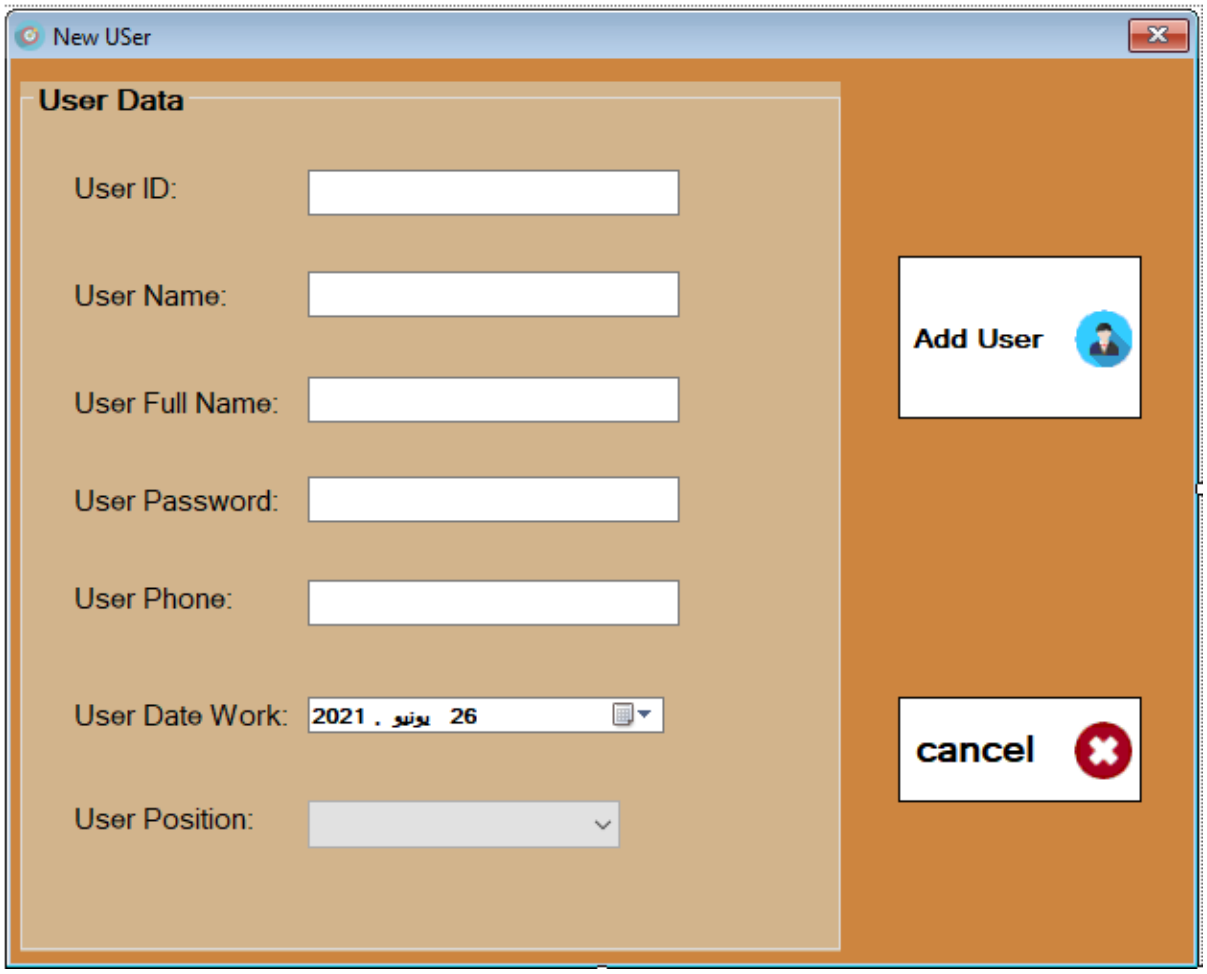
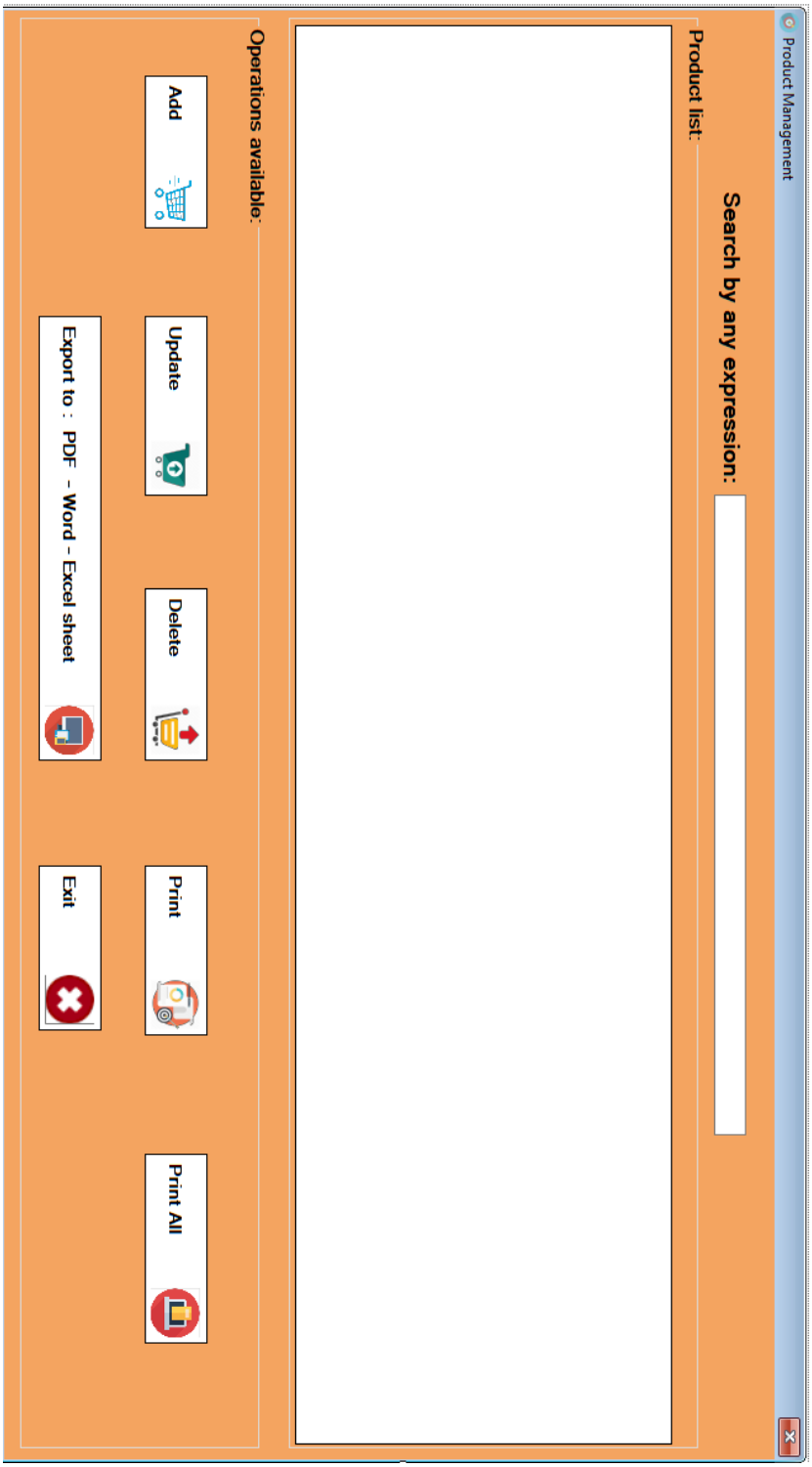


Figure 4.3: Sale operation

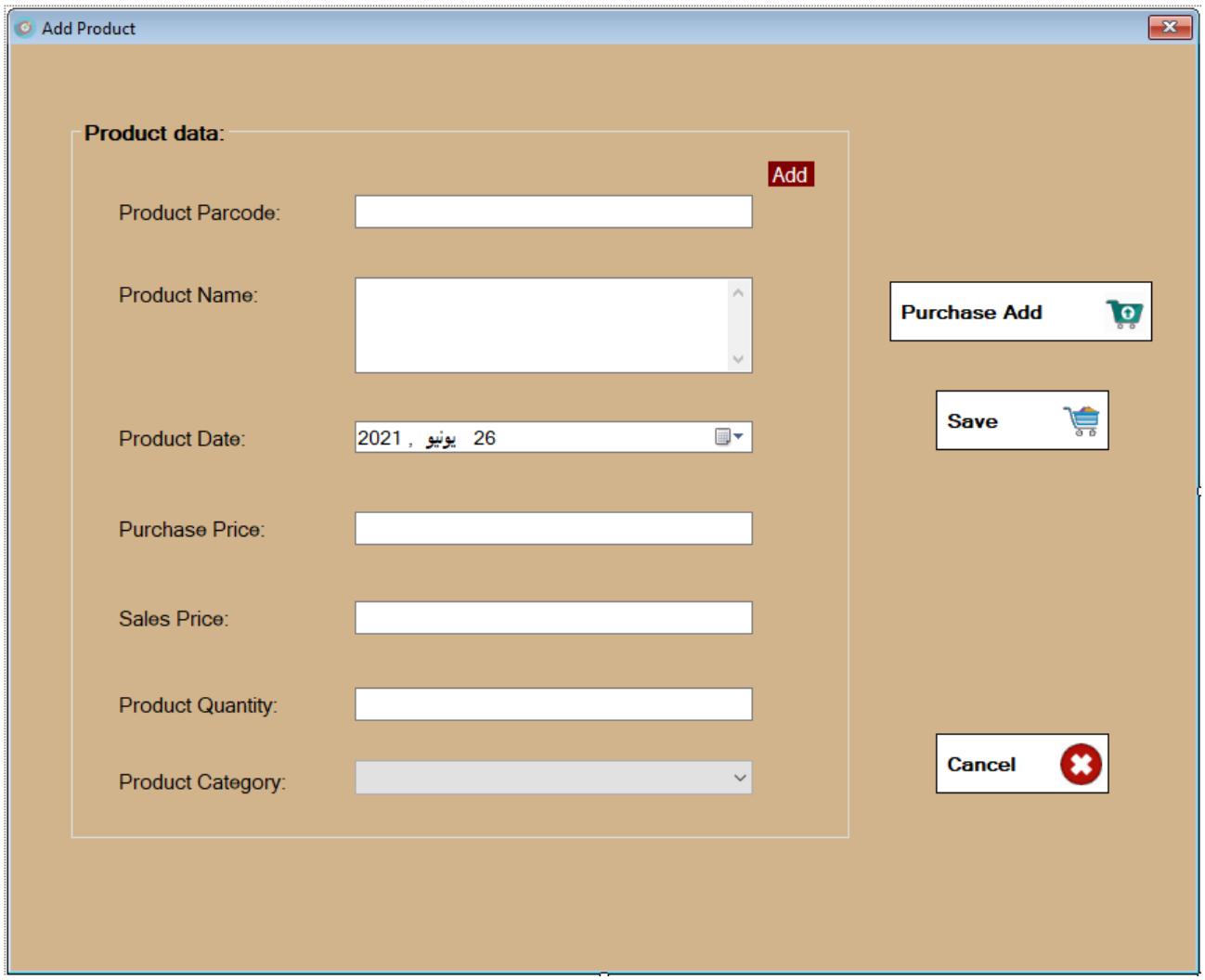
3.5 Add User form



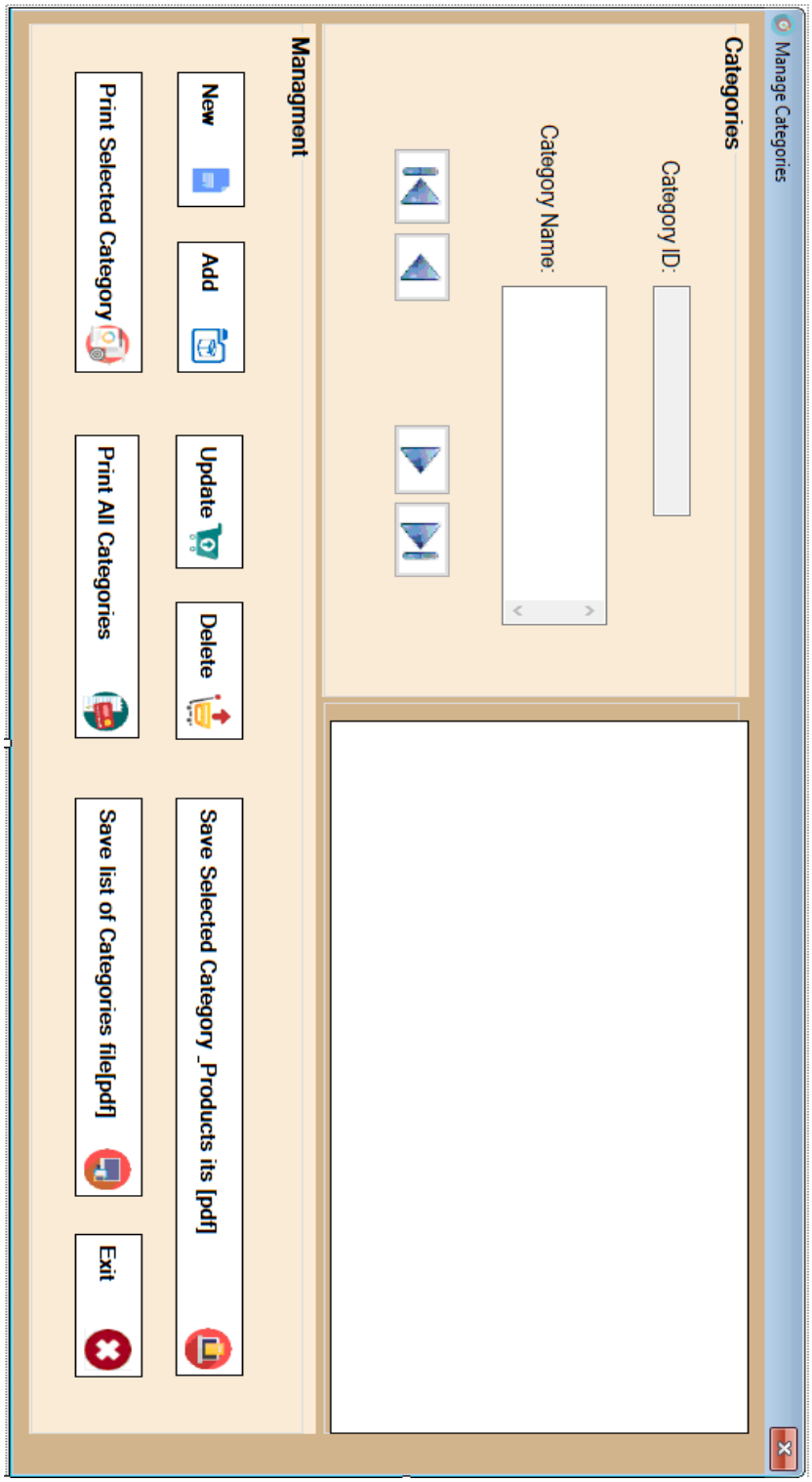
3.6 Product Management form



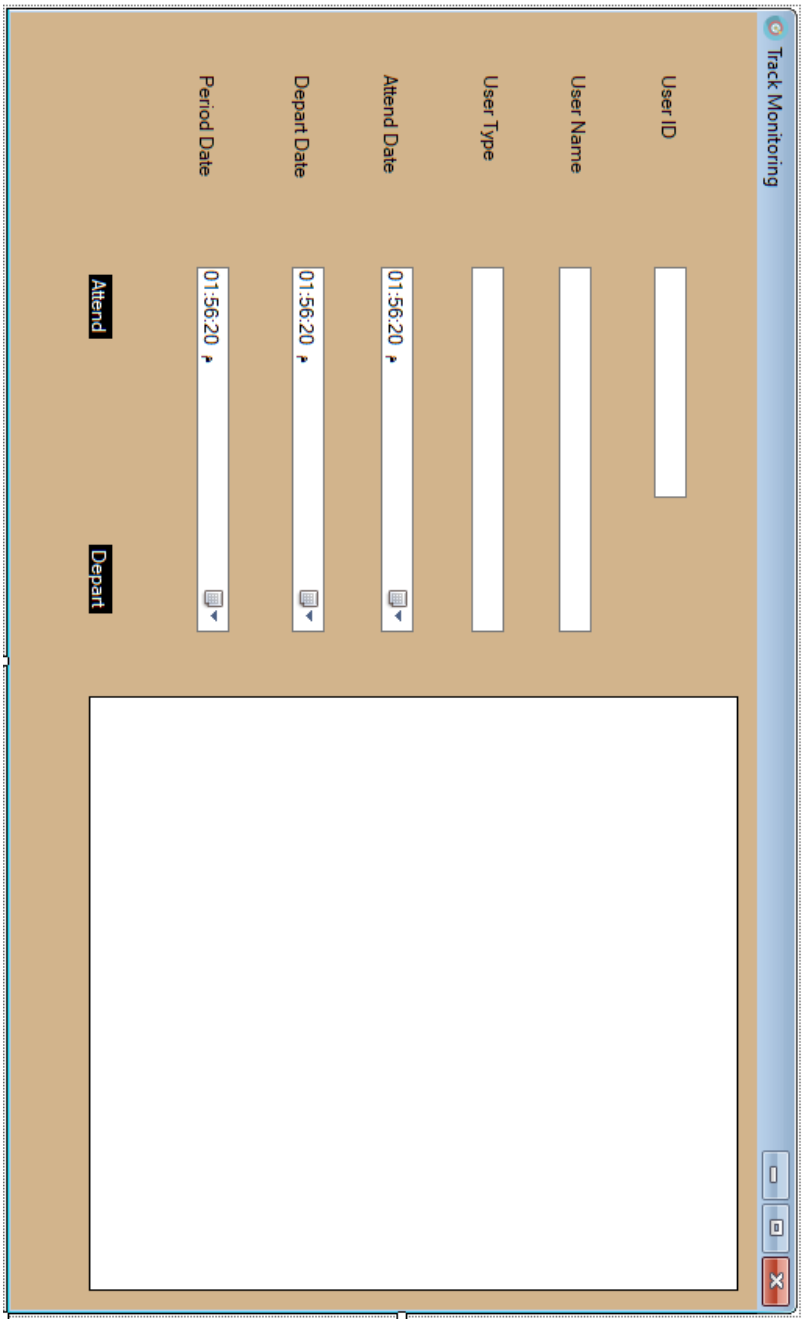
3.7 Add Product form



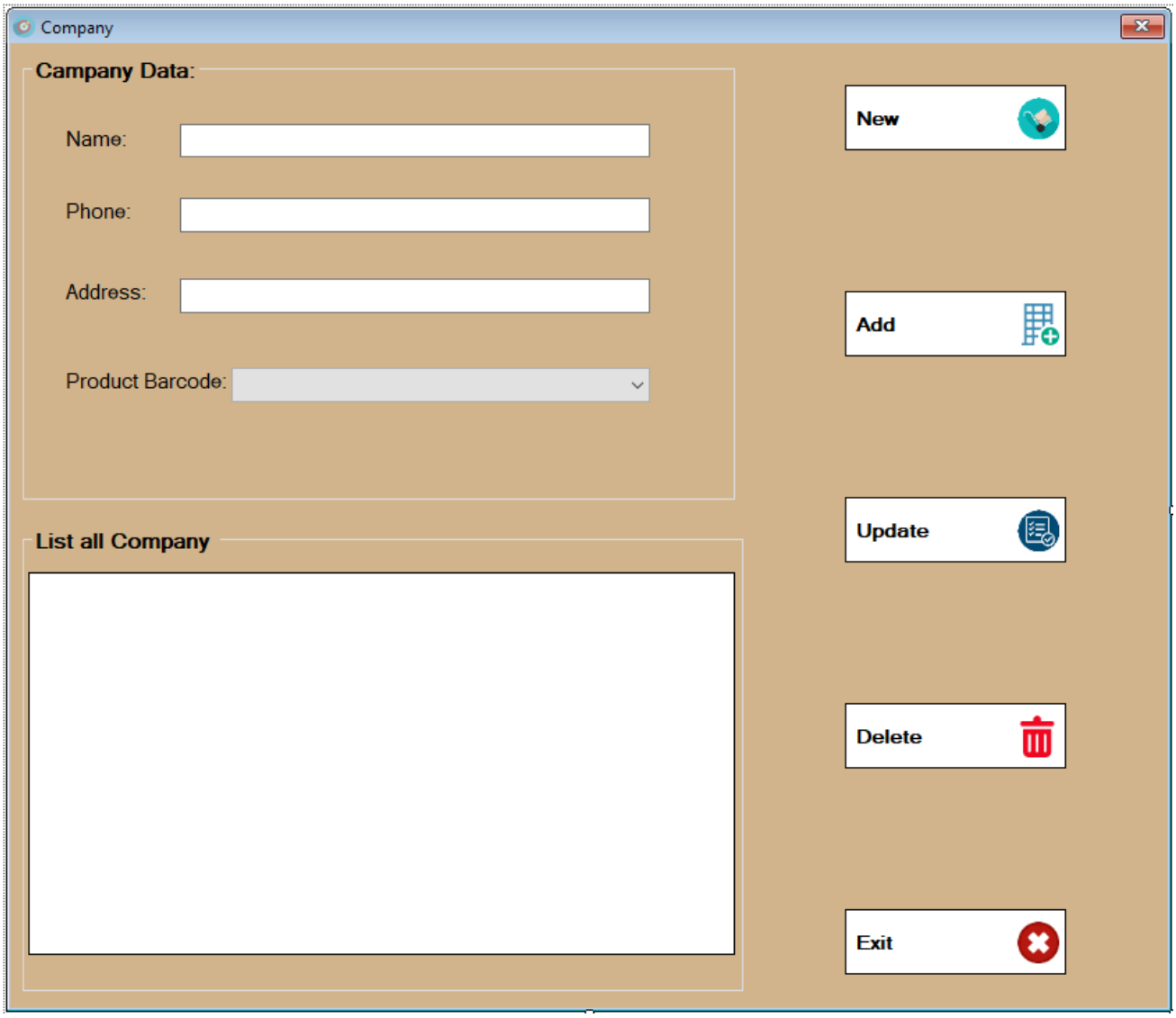
3.8 Manage Items form



3.9 Track monitoring form



3.10 Company form



Conclusion

The ‘SMMS is designed to provide a Windows based application that would make buy and sell, viewing and selection of a product easier. The search engine provides an easy and convenient way to search for products where an employee can Search for a product interactively and the search engine would refine the products available based on the user’s input. The user can then view the complete specification of each bill. They can also view the product quantity and also print their own reports. Use of MSSQL components would make the application interactive and prevents annoying post backs. Its GUI feature would make it easy to use.

Results & Challenges

The application can be used for any Windows application. It is easy to use, since it uses the GUI provided in the user dialog. User friendly screens are provided. The application is easy to use and interactive making online shopping a recreational activity for users. It has been thoroughly tested and implemented.

7.1 Challenges

- Compatibility with browsers like Mozilla Firefox, Internet explorer, .etc.
- Using a layered approach in developing the application which would make the application maintainable.
- Learning new technologies like using Visual C# for drag and drop tools and MS Sql database with high guidance. The overall idea of doing this project is to get a real time experience. Learn new technologies.

Limitations

This application does not have a built in check out process. An external checkout package has to be integrated in to this application. Also users cannot save the shift log transactions so that they can access later i.e. they cannot create wish items which they can access later. This application does not have features by which user can set quantity ranges for items and receive alerts once the quantity reaches the particular range.

Scope for Future Work

The following things can be done in future.

- The current system can be extended to allow the users to create shift log transactions and save products in to wish list.
- The users could subscribe for quantity alerts which would enable them to receive messages when quantity for products fall below a particular level.
- The current system is confined only to the mini and medium Supermarket process. It can be extended to have an easy to use check out process.
- Customers can have multiple billing information saved. During checkout they can use new feature to select billing information.
- The current system can be extended to integrate with online webpage and mobile application.

References

All about Microsoft controls in C#

http://www.msdn.microsoft.com/ , last Access: 2021/5/20.

- **Wikipedia for various diagrams & testing methods**

http://www.wikipedia.org/ , last Access: 2020/11/19.

- **K-State Research Exchange for samples in report writing**

http://krex.k-state.edu/dspace/handle/2097/959 , last Access: 2021/5/15.

- **Smart Draw for drawing all the Diagrams used in this report.**

http://www.smartdraw.com/ , last Access: 2020/11/22.

- **Sample Ecommerce Application**

http://www.NewEgg.com , last Access: 2020/12/9.

- Y.She, “The Design of small and medium-Sized Supermarket Management System,” Journal of Silicon valley vol10 no.12.
- H.Gaw, “The analysis and design of supermarket management system based on Membership” vol2.

Main Form

```
namespace smms
{
    public partial class Form_main : Form
    {
        private static Form_main frm;
        static void frm_formclosed(object sender, FormClosedEventArgs e)
        {
            frm = null;
        }
        public static Form_main getmainform
        {
            get
            {
                if (frm == null)
                {
                    frm = new Form_main();
                    frm.FormClosed += new FormClosedEventHandler(frm_formclosed);
                }
                return frm;
            }
        }

        public Form_main()
        {
            InitializeComponent();

            if(frm==null)   frm=this;
            this.productsToolStripMenuItem.Enabled = false;
            this.customersToolStripMenuItem.Enabled = false;
            this.userToolStripMenuItem.Enabled = false;
            this.createABackupToolStripMenuItem.Enabled = false;
            this.restoreABackupToolStripMenuItem.Enabled = false;
            this.supplierToolStripMenuItem.Enabled = false;
            this.signOutToolStripMenuItem.Enabled = false;
            this.bills.Enabled = false;
        }

        private void customerManagementToolStripMenuItem_Click(object sender, EventArgs e)
        {
            FORM_CUSTOMERS frm = new FORM_CUSTOMERS();
            frm.ShowDialog();
        }

        private void loginToolStripMenuItem_Click(object sender, EventArgs e)
        {
            login frm = new login();
            frm.ShowDialog();
        }




        private void addANewProductToolStripMenuItem_Click(object sender, EventArgs e)
        {
            FORM_ADD_PRODUCT frm = new FORM_ADD_PRODUCT();
            frm.ShowDialog();
        }

        private void productManagementToolStripMenuItem_Click(object sender, EventArgs e)
```

```
{
    FORM_PRODUCTS frm = new FORM_PRODUCTS();
    frm.ShowDialog();
}
private void addNewSuppliersToolStripMenuItem_Click(object sender, EventArgs e)
{
    supplier frm = new supplier();
    frm.ShowDialog();
}

private void suppliersManagementToolStripMenuItem_Click(object sender, EventArgs e)
{
    supplier_management frmsupman = new supplier_management();
    frmsupman.ShowDialog();
}
private void signOutToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form_main.getmainform.productsToolStripMenuItem.Enabled = false;
    Form_main.getmainform.customersToolStripMenuItem.Enabled = false;
    Form_main.getmainform.userToolStripMenuItem.Enabled = false;
    Form_main.getmainform.createABackupToolStripMenuItem.Enabled = false;
    Form_main.getmainform.restoreABackupToolStripMenuItem.Enabled = false;
    Form_main.getmainform.supplierToolStripMenuItem.Enabled = false;
    Form_main.getmainform.loginToolStripMenuItem.Enabled = true;
    Form_main.getmainform.signOutToolStripMenuItem.Enabled = false;
    Form_main.getmainform.bills.Enabled = false;
    MessageBox.Show("SignOut Done Succsfully" , "SMMS");
}
private void categoryManagementToolStripMenuItem_Click(object sender, EventArgs e)
{
    FORM_CATEGORIES frm = new FORM_CATEGORIES();
    frm.ShowDialog();
}

private void addNewSaleToolStripMenuItem_Click(object sender, EventArgs e)
{
    FORM_ORDER frm = new FORM_ORDER();
    frm.ShowDialog();
}
private void campaniesManagementToolStripMenuItem1_Click(object sender, EventArgs e)
{
    FORM_COMPANY frm = new FORM_COMPANY();
    frm.ShowDialog();
}

private void placesToolStripMenuItem_Click(object sender, EventArgs e)
{
    FORMPLACE frm = new FORMPLACE();
    frm.ShowDialog();
}

private void addToolStripMenuItem_Click(object sender, EventArgs e)
{
    FORM_USER frm = new FORM_USER();
    frm.ShowDialog();
}

private void userManagementToolStripMenuItem_Click(object sender, EventArgs e)
{
    FORM_USER_MANAGEMENT frm = new FORM_USER_MANAGEMENT();
    frm.ShowDialog();
```

```
        }

        private void purchasesManagementToolStripMenuItem_Click(object sender, EventArgs e)
        {
            FORM_PURCHASE_MANGER frm = new FORM_PURCHASE_MANGER();
            frm.ShowDialog();
        }

        private void addNewPurchaseToolStripMenuItem_Click(object sender, EventArgs e)
        {
            FORM_PURCHASE_BILL frm = new FORM_PURCHASE_BILL();
            frm.ShowDialog();
        }

        private void slesManagementToolStripMenuItem_Click(object sender, EventArgs e)
        {
            sale_management form = new sale_management();
            form.ShowDialog();
        }


        private void addReturnBillToolStripMenuItem_Click(object sender, EventArgs e)
        {
            FORM_RETURN_ORDER frm = new FORM_RETURN_ORDER();
            frm.ShowDialog();
        }

        private void returnBillManagementToolStripMenuItem_Click(object sender, EventArgs e)
        {
            FORM_RETURN_MANAGEMENT frm = new FORM_RETURN_MANAGEMENT();
            frm.ShowDialog();
        }
    }
}
```

Login Form

```
namespace smms{
    public partial class login : Form{
        BL.CLS_login log = new BL.CLS_login();
        public login(){InitializeComponent();}
        private void btncancel_Click(object sender, EventArgs e) {Close();}
        private void btnlogin_Click(object sender, EventArgs e) {
            if (txtUN.Text == string.Empty || txtPWD.Text == string.Empty) {
                MessageBox.Show("Please Rewrite user name or password!", "Attention",
MessageBoxButtons.OK, MessageBoxIcon.Information); return; }
            DataTable Dt = log.LOGIN(txtUN.Text, txtPWD.Text);
            if (Dt.Rows.Count > 0) {
                if (Dt.Rows[0][5].ToString() == "Manager"){
                    Form_main.getmainform.productsToolStripMenuItem.Enabled = true;
                    Form_main.getmainform.customersToolStripMenuItem.Enabled = true;
                    Form_main.getmainform.userToolStripMenuItem.Enabled = true;
                    Form_main.getmainform.createABackupToolStripMenuItem.Enabled = true;
                    Form_main.getmainform.restoreABackupToolStripMenuItem.Enabled = true;
                    Form_main.getmainform.supplierToolStripMenuItem.Enabled = true;
                    Form_main.getmainform.loginToolStripMenuItem.Enabled = false;
                    Form_main.getmainform.signOutToolStripMenuItem.Enabled = true;
                    Form_main.getmainform.bills.Enabled = true;
                    MessageBox.Show("Welcome " + Dt.Rows[0][1].ToString());
                    Program.SalesMan = Dt.Rows[0]["UserFullName"].ToString();
                    this.Close();
                }
                else if (Dt.Rows[0][5].ToString() == "The Data Officer"){
                    Form_main.getmainform.productsToolStripMenuItem.Enabled = true;
                    Form_main.getmainform.customersToolStripMenuItem.Enabled = false;
                    Form_main.getmainform.userToolStripMenuItem.Enabled = false;
                    Form_main.getmainform.createABackupToolStripMenuItem.Enabled = true;
                    Form_main.getmainform.restoreABackupToolStripMenuItem.Enabled = true;
                    Form_main.getmainform.supplierToolStripMenuItem.Enabled = true;
                    Form_main.getmainform.loginToolStripMenuItem.Enabled = false;
                    Form_main.getmainform.signOutToolStripMenuItem.Enabled = true;
                    Form_main.getmainform.bills.Enabled = false;
                    MessageBox.Show("Welcome " + Dt.Rows[0][1].ToString());
                    Program.SalesMan = Dt.Rows[0]["UserFullName"].ToString();
                    this.Close();
                }
                else if (Dt.Rows[0][5].ToString() == "Casher"){
                    Form_main.getmainform.productsToolStripMenuItem.Enabled = false;
                    Form_main.getmainform.customersToolStripMenuItem.Enabled = true;
                    Form_main.getmainform.userToolStripMenuItem.Enabled = false;
                    Form_main.getmainform.createABackupToolStripMenuItem.Enabled = false;
                    Form_main.getmainform.restoreABackupToolStripMenuItem.Enabled = false;
                    Form_main.getmainform.supplierToolStripMenuItem.Enabled = false;
                    Form_main.getmainform.loginToolStripMenuItem.Enabled = false;
                    Form_main.getmainform.signOutToolStripMenuItem.Enabled = true;
                    Form_main.getmainform.bills.Enabled = true;
                    Form_main.getmainform.addNewPurchaseToolStripMenuItem.Enabled = false;
                    Form_main.getmainform.purchasesManagementToolStripMenuItem.Enabled = false;
                    MessageBox.Show("Welcome " + Dt.Rows[0][1].ToString());
                    Program.SalesMan = Dt.Rows[0]["UserFullName"].ToString();
                    this.Close();}}}
        private void txtUN_KeyDown(object sender, KeyEventArgs e)
        {if (e.KeyCode == Keys.Enter) {txtPWD.Focus();}}

        private void txtPWD_KeyDown(object sender, KeyEventArgs e)
        {if (e.KeyCode == Keys.Enter) {btnlogin.Focus();}}}
```

Sale Operation form

```
namespace smms{
    public partial class FORM_ORDER : Form{
        DAL.DataAccessLayer dal = new DAL.DataAccessLayer();
        BL.CLS_ORDERS order = new BL.CLS_ORDERS();
        PL.FORM_PRODUCTS_LIST P_list = new FORM_PRODUCTS_LIST();
        DataTable Dt = new DataTable();

        void CalculateAmount()
        {
            if (text_prd_price.Text != string.Empty && text_prd_quantity.Text != string.Empty) {
text_prd_amount.Text = (Convert.ToDouble(text_prd_price.Text) *
Convert.ToInt32(text_prd_quantity.Text)).ToString();
            }
            else { text_prd_amount.Text = string.Empty; }
        }

        void CalculateTotalAmount()
        {
            if (text_prd_discounit.Text != string.Empty && text_prd_amount.Text != string.Empty)
            {
                double Discount = Convert.ToDouble(text_prd_discounit.Text);
                double Amount = Convert.ToDouble(text_prd_amount.Text);
                double TotalAmount = Amount - (Amount * (Discount / 100));

                text_prd_T_amount.Text = TotalAmount.ToString();
            }
            else { text_prd_T_amount.Text = string.Empty; }
        }

        void ClearBoxes()
        {
            text_prd_id.Clear();
            text_prd_name.Clear();
            text_prd_price.Clear();
            text_prd_quantity.Clear();
            text_prd_amount.Clear();
            text_prd_discounit.Clear();
            text_prd_T_amount.Clear();
            text_prd_id.Focus();
        }

        void ClearData()
        {
            text_order_id.Clear();
            text_order_des.Clear();
            dt_order.ResetText();
            text_cust_id.Clear();
            text_Cust_name.Clear();
            text_phone.Clear();
            text_address.Clear();
            text_Sum_Total.Clear();
            Dt.Clear();
            ClearBoxes();
            DGV_PRD_BILL.DataSource = null;
            pbox.Image = null;
            button_new.Enabled = false;
            radioButton_public.Enabled = true;
            radioButton_special.Enabled = true;
            button_Product.Enabled = false;
            text_prd_id.Enabled = false;
```

```
        }

        void ShowData()
        {
            Dt.Columns.Add("Product ID");
            Dt.Columns.Add("Product Name");
            Dt.Columns.Add("Price");
            Dt.Columns.Add("Quantity");
            Dt.Columns.Add("Amount");
            Dt.Columns.Add("Dscounit(%)");
            Dt.Columns.Add("Total Amount");

            DGV_PRD_BILL.DataSource = Dt;
        }


        public FORM_ORDER()
        {
            InitializeComponent();

            ShowData();

            button_new.Enabled = false;
            radioButton_public.Enabled = true;
            radioButton_special.Enabled = true;
            button_Product.Enabled = false;
            text_prd_id.Enabled = false;
            text_sales_man.Text = Program.SalesMan;
        }

        private void FORM_ORDER_Load(object sender, EventArgs e){radioButton_public.Focus();}

        private void button3_Click(object sender, EventArgs e)
        {
            this.text_order_id.Text = order.GET_LAST_BILL_ID().Rows[0][0].ToString();
            button_new.Enabled = false;
            butto_add.Enabled = true;
            button_Product.Enabled = true;
            text_order_des.Text = "Bill TO " + text_Cust_name.Text;
            text_prd_id.Enabled = true;
            text_prd_id.Focus();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            try
            {
                order.ADD_ORDERS(Convert.ToInt32(text_order_id.Text), dt_order.Value,
text_order_des.Text, Convert.ToInt32(text_cust_id.Text), text_sales_man.Text);
                for (int i = 0; i < DGV_PRD_BILL.Rows.Count - 1; i++)
                {
                    order.ADD_ORDERS_DETAILS(DGV_PRD_BILL.Rows[i].Cells[0].Value.ToString(),
Convert.ToInt32(text_order_id.Text)
                        , DGV_PRD_BILL.Rows[i].Cells[1].Value.ToString(),
DGV_PRD_BILL.Rows[i].Cells[2].Value.ToString()
                        , Convert.ToInt32(DGV_PRD_BILL.CurrentRow.Cells[3].Value),
Convert.ToInt32(DGV_PRD_BILL.Rows[i].Cells[5].Value)
                        , DGV_PRD_BILL.Rows[i].Cells[4].Value.ToString(),
DGV_PRD_BILL.Rows[i].Cells[6].Value.ToString());
                }
```

```
                MessageBox.Show("Saved Successfully", "Save Operation", MessageBoxButtons.OK,
MessageBoxIcon.Information);

                if (MessageBox.Show("Do you want to Print this Order?", "Inquiry Process",
MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation) == DialogResult.Yes)
                {

                    button_print.Enabled = true;
                    butto_add.Enabled = false;
                }
                else
                {
                    MessageBox.Show("Print Canceled", "Print Process", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
                    button_new.Enabled = true;
                }
            }
            catch
            { return; }
        }

        private void button5_Click(object sender, EventArgs e) {this.Close();}

        private void button3_Click_1(object sender, EventArgs e)
        {
            ClearBoxes();
            FORM_PRODUCTS_LIST form = new FORM_PRODUCTS_LIST();
            form.ShowDialog();
            text_prd_id.Text = form.DGV_PRODUCTS.CurrentRow.Cells[0].Value.ToString();
            text_prd_name.Text = form.DGV_PRODUCTS.CurrentRow.Cells[1].Value.ToString();
            text_prd_price.Text = form.DGV_PRODUCTS.CurrentRow.Cells[3].Value.ToString();
            text_prd_quantity.Focus();

        }

        private void text_prd_quantity_KeyPress(object sender, KeyPressEventArgs e)
        {
            if (!char.IsDigit(e.KeyChar) && e.KeyChar != 8)
            {
                e.Handled = true;
            }
        }
        private void text_prd_price_KeyPress(object sender, KeyPressEventArgs e)
        {
            char DecimalSeparator =
Convert.ToChar(CultureInfo.CurrentCulture.NumberFormat.NumberDecimalSeparator);

            if (!char.IsDigit(e.KeyChar) && e.KeyChar != 8 && e.KeyChar != DecimalSeparator)
            {
                e.Handled = true;
            }
        }

        private void text_prd_price_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyCode == Keys.Enter && text_prd_price.Text != string.Empty)
            {
                text_prd_quantity.Focus();
            }
        }
```

```
        private void text_prd_quantity_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyCode == Keys.Enter)
            {
                if (text_prd_quantity.Text != string.Empty)
                {
                    if (order.VerifyProductQte(text_prd_id.Text,
Convert.ToInt32(text_prd_quantity.Text)).Rows.Count < 1)
                    {
                        MessageBox.Show("The Input Quantity Not Available !", "Attention",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                        text_prd_quantity.Focus();
                        text_prd_quantity.SelectionStart = 0;
                        text_prd_quantity.SelectionLength = text_prd_quantity.TextLength;
                        return;
                    }

                    for (int i = 0; i < DGV_PRD_BILL.Rows.Count - 1; i++)
                    {
                        if (DGV_PRD_BILL.Rows[i].Cells[0].Value.ToString() == text_prd_id.Text)
                        {
                            if (MessageBox.Show("Do you want to Edit Quantity the Entered product?",
"Edit process", MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation) == DialogResult.Yes)
                            {
                                MessageBox.Show("Successfully Edited", "Editing process",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                                DGV_PRD_BILL.Rows.RemoveAt(DGV_PRD_BILL.Rows[i].Index);
                            }
                            else
                            {
                                MessageBox.Show("Edit Canceled", "Editing Process",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                            }
                        }
                    }

                }
                else { return; }
            }

            if (e.KeyCode == Keys.Enter && text_prd_quantity.Text != string.Empty)
            {
                text_prd_discounit.Focus();
            }

        }

        private void text_prd_price_KeyUp(object sender, KeyEventArgs e)
        {
            CalculateAmount();
            CalculateTotalAmount();
        }

        private void text_prd_quantity_KeyUp(object sender, KeyEventArgs e)
        {
            CalculateAmount();
            CalculateTotalAmount();
        }

        private void text_prd_discounit_KeyPress(object sender, KeyPressEventArgs e)
        {
```

```
            char DecimalSeparator =
Convert.ToChar(CultureInfo.CurrentCulture.NumberFormat.NumberDecimalSeparator);

            if (!char.IsDigit(e.KeyChar) && e.KeyChar != 8 && e.KeyChar != DecimalSeparator)
            {
                e.Handled = true;
            }
        }

        private void text_prd_discounit_KeyUp(object sender, KeyEventArgs e)
        {
            CalculateTotalAmount();
        }

        private void text_prd_discounit_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyCode == Keys.Enter)
            {
                DataRow r = Dt.NewRow();
                r[0] = text_prd_id.Text;
                r[1] = text_prd_name.Text;
                r[2] = text_prd_price.Text;
                r[3] = text_prd_quantity.Text;
                r[4] = text_prd_amount.Text;
                r[5] = text_prd_discounit.Text;
                r[6] = text_prd_T_amount.Text;

                Dt.Rows.Add(r);
                DGV_PRD_BILL.DataSource = Dt;
                ClearBoxes();
                text_Sum_Total.Text =
                    (from DataGridViewRow row in DGV_PRD_BILL.Rows
                     where row.Cells[6].FormattedValue.ToString() != string.Empty
                     select Convert.ToDouble(row.Cells[6].FormattedValue)).Sum().ToString();
            }
        }

        private void DGV_PRD_BILL_DoubleClick(object sender, EventArgs e)
        {
            try
            {
                text_prd_id.Text = this.DGV_PRD_BILL.CurrentRow.Cells[0].Value.ToString();
                text_prd_name.Text = this.DGV_PRD_BILL.CurrentRow.Cells[1].Value.ToString();
                text_prd_price.Text = this.DGV_PRD_BILL.CurrentRow.Cells[2].Value.ToString();
                text_prd_quantity.Text = this.DGV_PRD_BILL.CurrentRow.Cells[3].Value.ToString();
                text_prd_amount.Text = this.DGV_PRD_BILL.CurrentRow.Cells[4].Value.ToString();
                text_prd_discounit.Text = this.DGV_PRD_BILL.CurrentRow.Cells[5].Value.ToString();
                text_prd_T_amount.Text = this.DGV_PRD_BILL.CurrentRow.Cells[6].Value.ToString();

                DGV_PRD_BILL.Rows.RemoveAt(DGV_PRD_BILL.CurrentRow.Index);
                text_prd_quantity.Focus();
            }
            catch { return; }
        }

        private void DGV_PRD_BILL_RowsRemoved(object sender, DataGridViewRowsRemovedEventArgs e)
        {
            text_Sum_Total.Text =
                    (from DataGridViewRow row in DGV_PRD_BILL.Rows
                     where row.Cells[6].FormattedValue.ToString() != string.Empty
                     select Convert.ToDouble(row.Cells[6].FormattedValue)).Sum().ToString();
```

```
        }

        private void editToolStripMenuItem_Click(object sender, EventArgs e)
        {
            DGV_PRD_BILL_DoubleClick(sender, e);
        }

        private void deleteCurrentRowToolStripMenuItem_Click(object sender, EventArgs e)
        {
            DGV_PRD_BILL.Rows.RemoveAt(DGV_PRD_BILL.CurrentRow.Index);
        }

        private void deleteTheAllToolStripMenuItem_Click(object sender, EventArgs e)
        {
            Dt.Clear();
            DGV_PRD_BILL.Refresh();
        }

        private void text_prd_id_TextChanged(object sender, EventArgs e)
        {
            try
            {
                if (text_prd_id.Text != string.Empty)
                {
                    DataTable dt = new DataTable();
                    dt = order.GetproductBarcode(text_prd_id.Text);
                    if (dt.Rows.Count > 0)
                    {
                        text_prd_id.Text = dt.Rows[0][0].ToString();
                        text_prd_name.Text = dt.Rows[0][1].ToString();
                        text_prd_price.Text = dt.Rows[0][3].ToString();
                    }
                }
            }
            catch { return; }
        }

        private void radioButton1_CheckedChanged(object sender, EventArgs e)
        {
            text_cust_id.Clear();
            text_Cust_name.Clear();
            text_phone.Clear();
            text_address.Clear();
            pbox.Image = null;
            panelCus.Enabled = false;
            button_new.Enabled = true;
            radioButton_special.Checked = false;
            panelCus.Enabled = false;
            text_cust_id.Text = "1040";
            text_Cust_name.Text = "Public Customer";

        }

        private void radioButton_special_CheckedChanged(object sender, EventArgs e)
        {
            text_cust_id.Clear();
            text_Cust_name.Clear();
            text_phone.Clear();
            text_address.Clear();
            pbox.Image = null;
            panelCus.Enabled = true;
```

```
        }

        private void radioButton_public_Click(object sender, EventArgs e)
        {
            button_new.Enabled = true;
            radioButton_special.Checked = false;
            panelCus.Enabled = false;
            text_cust_id.Text = "1040";
            text_Cust_name.Text = "Public Customer";
            text_phone.Clear();
            text_address.Clear();
        }

        private void radioButton_special_Click(object sender, EventArgs e)
        {
            pbox.Image = null;
            FORM_CUSTOMERS_LIST form = new FORM_CUSTOMERS_LIST();
            form.ShowDialog();

            if (form.DGV_Customers.CurrentRow.Cells[6].Value is DBNull)
            {

                this.text_cust_id.Text = form.DGV_Customers.CurrentRow.Cells[0].Value.ToString();
                this.text_Cust_name.Text = form.DGV_Customers.CurrentRow.Cells[1].Value.ToString();
                this.text_phone.Text = form.DGV_Customers.CurrentRow.Cells[2].Value.ToString();
                this.text_address.Text = form.DGV_Customers.CurrentRow.Cells[4].Value.ToString();
                MessageBox.Show("This client does not have an image");

            }
            else
            {
                this.text_cust_id.Text = form.DGV_Customers.CurrentRow.Cells[0].Value.ToString();
                this.text_Cust_name.Text = form.DGV_Customers.CurrentRow.Cells[1].Value.ToString();
                this.text_phone.Text = form.DGV_Customers.CurrentRow.Cells[2].Value.ToString();
                this.text_address.Text = form.DGV_Customers.CurrentRow.Cells[4].Value.ToString();

                byte[] custpicture = (byte[])form.DGV_Customers.CurrentRow.Cells[6].Value;
                MemoryStream ms = new MemoryStream(custpicture);
                this.pbox.Image = Image.FromStream(ms);
            }

            radioButton_public.Checked = false;
            panelCus.Enabled = true;
        }
        private void button_print_Click_1(object sender, EventArgs e)
        {
            int Order_ID = Convert.ToInt32(order.GET_last_ORDER().Rows[0][0]);
            RPT.RPT_ORDERS Report = new RPT.RPT_ORDERS();
            RPT.FRM_RPT_PRODUCT frm = new RPT.FRM_RPT_PRODUCT();
            Report.SetDataSource(order.GET_ORDER_DETAILS(Order_ID));
            frm.crystalReportViewer1.ReportSource = Report;
            frm.ShowDialog();
        }
    }
}
```

Order list form

```
namespace smms
{
    public partial class FORM_R_ORDER_LIST : Form
    {
        BL.CLS_RETURN_ORDER R_ORDER = new BL.CLS_RETURN_ORDER();
        public FORM_R_ORDER_LIST()
        {
            InitializeComponent();
            DGV_R_ORDER.DataSource = R_ORDER.GET_R_order();
        }

        private void DGV_R_ORDER_DoubleClick(object sender, EventArgs e)
        {

            FORM_RETURN_ORDER.getmainform.txtorderid.Text =
                                            DGV_R_ORDER.CurrentRow.Cells[1].Value.ToString();
            FORM_RETURN_ORDER.getmainform.Barcode.Text =
                                            DGV_R_ORDER.CurrentRow.Cells[0].Value.ToString();
            FORM_RETURN_ORDER.getmainform.Product_name.Text =
                                            DGV_R_ORDER.CurrentRow.Cells[2].Value.ToString();
            FORM_RETURN_ORDER.getmainform.R_price.Text =
                                            DGV_R_ORDER.CurrentRow.Cells[3].Value.ToString();
            FORM_RETURN_ORDER.getmainform.Qte.Text =
                                            DGV_R_ORDER.CurrentRow.Cells[4].Value.ToString();
            FORM_RETURN_ORDER.getmainform.Dis.Text =
                                            DGV_R_ORDER.CurrentRow.Cells[5].Value.ToString();
            FORM_RETURN_ORDER.getmainform.T_Pric.Text =
                                            DGV_R_ORDER.CurrentRow.Cells[7].Value.ToString();

            this.Close();

        }
    }
}
```

User form

```
namespace smms.PL
{
    public partial class FORM_USER : Form
    {
        private static FORM_USER frm;
        static void frm_formclosed(object sender, FormClosedEventArgs e) {frm = null; }
        public static FORM_USER getmainform
        {
            get
            {
                if (frm == null)
                {
                    frm = new FORM_USER();
                    frm.FormClosed += new FormClosedEventHandler(frm_formclosed);
                }
                return frm;
            }
        }
        BL.CLS_USER clauser = new BL.CLS_USER();
        public string state = "Add";
        public FORM_USER()
        {
            InitializeComponent();
            if (frm == null) frm = this;
        }

        void clear()
        {
            dateTimePicker1.ResetText();
            text_id_user.Clear();
            text_name_user.Clear();
            text_PW_user.Clear();
            text_Fname_user.Clear();
            text_PH_user.Clear();
            com_Position_user.SelectedIndex = -1;
        }

        private void button2_Click(object sender, EventArgs e)
        {
            if (state == "Add") Close();
            else
            {
                Close();
                FORM_USER_MANAGEMENT.getmainform.btn_add_user.Enabled = true;
                FORM_USER_MANAGEMENT.getmainform.btn_delete.Enabled = false;
                FORM_USER_MANAGEMENT.getmainform.btn_update.Enabled = false;
            }
        }
        private void btn_add_Click(object sender, EventArgs e)
        {
            if (state == "Add")
            {
                if (text_id_user.Text == string.Empty || text_name_user.Text == string.Empty ||
                                                         text_Fname_user.Text == string.Empty||
                    text_PW_user.Text == string.Empty || text_PH_user.Text == string.Empty ||
                                                         com_Position_user.Text == string.Empty)
                {
                    MessageBox.Show("Rewrite all data !", "Attention", MessageBoxButtons.OK,
                                                                       MessageBoxIcon.Information);
                    return;
```

```
            }
            clauser.AddUser(Convert.ToInt32(text_id_user.Text), text_name_user.Text,
                text_PW_user.Text, text_Fname_user.Text, dateTimePicker1.Value,
                com_Position_user.Text, Convert.ToInt32(text_PH_user.Text));
            MessageBox.Show("Add User Done Successfuly", "SMMS", MessageBoxButtons.OK,
                            MessageBoxIcon.Information);
            clear();
        }
        else
        {
            if (text_id_user.Text == string.Empty || text_name_user.Text == string.Empty ||
                                                     text_Fname_user.Text == string.Empty||
                text_PW_user.Text == string.Empty || text_PH_user.Text == string.Empty ||
                                                     com_Position_user.Text == string.Empty)
            {
                MessageBox.Show("Rewrite all data !", "Attention", MessageBoxButtons.OK,
                                                     MessageBoxIcon.Information);
                return;
            }
            clauser.Update_User(Convert.ToInt32(text_id_user.Text), text_name_user.Text,
                text_PW_user.Text, text_Fname_user.Text, dateTimePicker1.Value,
                com_Position_user.Text, Convert.ToInt32(text_PH_user.Text));
            MessageBox.Show("EDIT User Successfuly", "SMMS", MessageBoxButtons.OK,
                                                      MessageBoxIcon.Information);
            FORM_USER_MANAGEMENT.getmainform.btn_add_user.Enabled = true;
            FORM_USER_MANAGEMENT.getmainform.btn_delete.Enabled = false;
            FORM_USER_MANAGEMENT.getmainform.btn_update.Enabled = false;
            clear();
            this.Close();
        }
        FORM_USER_MANAGEMENT.getmainform.DGV_user.DataSource = clauser.GET_ALL_USER();
    }
    private void text_id_user_Validated(object sender, EventArgs e)
    {
        if (state == "Add")
        {
            DataTable Dt = new DataTable();
            Dt = clauser.VerifyUserID(text_id_user.Text);
            if (Dt.Rows.Count > 0)
            {
                MessageBox.Show("This identifier already exists ", "Warning!",
                                MessageBoxButtons.OK, MessageBoxIcon.Warning);
                text_id_user.Focus();
                text_id_user.SelectionStart = 0;
                text_id_user.SelectionLength = text_id_user.TextLength;
            }
        }
    }
    private void text_id_user_KeyDown(object sender, KeyEventArgs e)
    {if (e.KeyCode == Keys.Enter) text_name_user.Focus();}}
    private void text_name_user_KeyDown(object sender, KeyEventArgs e)
    {if (e.KeyCode == Keys.Enter) {text_Fname_user.Focus();}}
    private void text_Fname_user_KeyDown(object sender, KeyEventArgs e)
    {if (e.KeyCode == Keys.Enter) {text_PW_user.Focus();}}
    private void text_PW_user_KeyDown(object sender, KeyEventArgs e)
    {if (e.KeyCode == Keys.Enter) {text_PH_user.Focus();}}
    private void text_PH_user_KeyDown(object sender, KeyEventArgs e)
    {if (e.KeyCode == Keys.Enter) {dateTimePicker1.Focus();}}
    private void dateTimePicker1_KeyDown(object sender, KeyEventArgs e)
    {if (e.KeyCode == Keys.Enter) {com_Position_user.Focus();}}
    private void com_Position_user_KeyDown(object sender, KeyEventArgs e)
```

```
        {if (e.KeyCode == Keys.Enter) {btn_add.Focus();}}
    }
}
```

PRODUCTS Form

```
namespace smms
{
    public partial class FORM_PRODUCTS : Form
    {
        private static FORM_PRODUCTS frm;
        static void frm_formclosed(object sender, FormClosedEventArgs e)
        {frm = null;}

        public static FORM_PRODUCTS getmainform
        { if (frm == null)
                {
                    frm = new FORM_PRODUCTS();
                    frm.FormClosed += new FormClosedEventHandler(frm_formclosed);
                }
                return frm;
            }
        }

        BL.CLS_Product prd = new BL.CLS_Product();
        public FORM_PRODUCTS()
        {
            InitializeComponent();
            btn_update.Enabled = false;
            btn_delete.Enabled = false;
            btn_print.Enabled = false;
            btn_add.Enabled = true;

            if (frm == null) frm = this;
            this.dataGridView1.DataSource = prd.GET_ALL_PRODUCTS();
        }

        private void txtsearch_TextChanged(object sender, EventArgs e)
        {
            DataTable Dt = new DataTable();
            Dt = prd.SeachProduct(txtsearch.Text);
            this.dataGridView1.DataSource=Dt;
        }

        private void button1_Click(object sender, EventArgs e)
        {FORM_ADD_PRODUCT frm = new FORM_ADD_PRODUCT();frm.ShowDialog();frm.txtRef.Focus();}

        private void button2_Click(object sender, EventArgs e)
        {
            if (MessageBox.Show("Do you want to delete the selected product?", "Deletion process",
                MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation) == DialogResult.Yes)
            {
                prd.DeleteProduct(this.dataGridView1.CurrentRow.Cells[0].Value.ToString());
                MessageBox.Show("Successfully deleted", "Deletion process", MessageBoxButtons.OK,
                                                                    MessageBoxIcon.Information);
                this.dataGridView1.DataSource = prd.GET_ALL_PRODUCTS();
            }
            else
            {
```

```
            MessageBox.Show("Edit Canceled", "Edit Process", MessageBoxButtons.OK,
                                                             MessageBoxIcon.Exclamation);
        }
        btn_update.Enabled = false;
        btn_delete.Enabled = false;
        btn_print.Enabled = false;
        btn_add.Enabled = true;
    }

    private void button3_Click(object sender, EventArgs e)
    {
        FORM_ADD_PRODUCT form = new FORM_ADD_PRODUCT();
        form.button1.Visible = false;
        form.txtRef.Text=this.dataGridView1.CurrentRow.Cells[0].Value.ToString();
        form.txtDes.Text = this.dataGridView1.CurrentRow.Cells[1].Value.ToString();
        form.txtqte.Text = this.dataGridView1.CurrentRow.Cells[5].Value.ToString();
        form.txt_P_Price.Text = this.dataGridView1.CurrentRow.Cells[3].Value.ToString();
        form.txt_S_price.Text = this.dataGridView1.CurrentRow.Cells[4].Value.ToString();
        form.cmbCategories.Text = this.dataGridView1.CurrentRow.Cells[6].Value.ToString();
        form.Text = "Product Update:" + this.dataGridView1.CurrentRow.Cells[1].Value.ToString();
        form.btnAdd.Text = "Update";
        form.state = "update";
        form.txtRef.ReadOnly = true;
        form.ShowDialog();
    }

    private void button6_Click(object sender, EventArgs e) {this.Close();}

    private void button7_Click(object sender, EventArgs e)
    {
        RPT.RPT_ALL_PRODUCTS myreport = new RPT.RPT_ALL_PRODUCTS();
        ExportOptions export = new ExportOptions();
        DiskFileDestinationOptions dfdoptions = new DiskFileDestinationOptions();
        ExcelFormatOptions excelformat=new ExcelFormatOptions();
        dfdoptions.DiskFileName = "E:\\Productslist.xls";
        export = myreport.ExportOptions;
        export.ExportDestinationType = ExportDestinationType.DiskFile;
        export.ExportFormatType = ExportFormatType.Excel;
        export.ExportFormatOptions = excelformat;
        export.ExportDestinationOptions = dfdoptions;
        myreport.Export();
        MessageBox.Show("List Exported Successfuly ! ", "Export", MessageBoxButtons.OK,
                                                                  MessageBoxIcon.Information);
    }

    private void button4_Click(object sender, EventArgs e)
    {
        RPT.RPT_TO_SINGLE myreport = new RPT.RPT_TO_SINGLE();
        myreport.SetParameterValue("@ID", this.dataGridView1.CurrentRow.Cells[0].Value.ToString());
        RPT.FRM_RPT_PRODUCT MYFORM = new RPT.FRM_RPT_PRODUCT();
        MYFORM.crystalReportViewer1.ReportSource = myreport;
        MYFORM.ShowDialog();
        btn_update.Enabled = false;
        btn_delete.Enabled = false;
        btn_print.Enabled = false;
        btn_add.Enabled = true;
    }

    private void button5_Click(object sender, EventArgs e)
    {
        RPT.RPT_ALL_PRODUCTS myreport = new RPT.RPT_ALL_PRODUCTS();
```

```
            RPT.FRM_RPT_PRODUCT myform = new RPT.FRM_RPT_PRODUCT();
            myform.crystalReportViewer1.ReportSource = myreport;
            myform.ShowDialog();
        }

        private void dataGridView1_DoubleClick(object sender, EventArgs e)
        {btn_update.Enabled = true; btn_delete.Enabled = true; btn_print.Enabled = true;}
    }
}
```

ADD_PRODUCT Form

```
namespace smms
{
    public partial class FORM_ADD_PRODUCT : Form
    {
        BL.CLS_PURCHASE clsspur = new BL.CLS_PURCHASE();
        BL.CLS_Product clsspro = new BL.CLS_Product();
        private static FORM_ADD_PRODUCT frm;
        static void frm_formclosed(object sender, FormClosedEventArgs e)
        {frm = null; }

        public static FORM_ADD_PRODUCT getmainform
        {
            if (frm == null)
                {
                    frm = new FORM_ADD_PRODUCT();
                    frm.FormClosed += new FormClosedEventHandler(frm_formclosed);
                }
                return frm;
            }
        }
        public string state = "add";
        BL.CLS_Product prd = new BL.CLS_Product();
        public FORM_ADD_PRODUCT()
        {
            InitializeComponent();

            if (frm == null)
                frm = this;
            cmbCategories.DataSource = prd.GET_ALL_CATEGORIES();
            cmbCategories.DisplayMember = "CatName";
            cmbCategories.ValueMember = "CatID";
            cmbCategories.SelectedIndex = -1;
            labeladd.Visible = false;

        }
        void clear()
        {
            txtRef.Clear();
            txtDes.Clear();
            txt_P_Price.Clear();
            txt_S_price.Clear();
            txtqte.Clear();
            cmbCategories.SelectedIndex = -1;
        }
        private void btnlogin_Click(object sender, EventArgs e)
        {
            if (txtRef.Text == string.Empty || txtDes.Text == string.Empty || txt_P_Price.Text ==
```

```
                                                        string.Empty
            || txt_S_price.Text == string.Empty ||
               txtqte.Text == string.Empty||cmbCategories.Text==string.Empty)
        {
            MessageBox.Show("Please Rewrite All Data !", "Attention", MessageBoxButtons.OK,
                                                        MessageBoxIcon.Information);
            return;
        }
            if (state == "add")
            {
                prd.ADD_PRODUCTS(txtRef.Text, txtDes.Text, textDate.Value, txt_P_Price.Text,
                                 txt_S_price.Text, Convert.ToInt32(txtqte.Text),
                                 Convert.ToInt32(cmbCategories.SelectedValue));
                clsspro.UPDATE_STATE(txtRef.Text, labeladd.Text, txtDes.Text);
                MessageBox.Show("Successfully added", "Operation added", MessageBoxButtons.OK,
                                                        MessageBoxIcon.Information);
                clear();
            }
            else
            {
                prd.Update_PRODUCTS(txtRef.Text, txtDes.Text, textDate.Value, txt_P_Price.Text,
                                   txt_S_price.Text, Convert.ToInt32(txtqte.Text),
                                   Convert.ToInt32(cmbCategories.SelectedValue));
                clear();
                cmbCategories.SelectedIndex = -1;

                MessageBox.Show("Modified successfully", "Modification process",
                                        MessageBoxButtons.OK, MessageBoxIcon.Information);
                this.Close();
                FORM_PRODUCTS.getmainform.btn_update.Enabled = false;
                FORM_PRODUCTS.getmainform.btn_delete.Enabled = false;
                FORM_PRODUCTS.getmainform.btn_print.Enabled = false;
                FORM_PRODUCTS.getmainform.btn_add.Enabled = true;
            }
            FORM_PRODUCTS.getmainform.dataGridView1.DataSource = prd.GET_ALL_PRODUCTS
                                                                txtRef.Focus();
    }

    private void txtRef_Validated(object sender, EventArgs e)
    {
        if (state == "add")
        {
            DataTable Dt = new DataTable();
            Dt = prd.VerifyProductID(txtRef.Text);
            if (Dt.Rows.Count > 0)
            {
                MessageBox.Show("This identifier already exists ", "Warning!",
                                MessageBoxButtons.OK, MessageBoxIcon.Warning);
                txtRef.Focus();
                txtRef.SelectionStart = 0;
                txtRef.SelectionLength = txtRef.TextLength;
            }
        }
    }

    private void btncancel_Click(object sender, EventArgs e)
    {
        Close();
        FORM_PRODUCTS.getmainform.btn_update.Enabled = false;
        FORM_PRODUCTS.getmainform.btn_delete.Enabled = false;
        FORM_PRODUCTS.getmainform.btn_print.Enabled = false;
```

```
            FORM_PRODUCTS.getmainform.btn_add.Enabled = true;
        }

        private void txtRef_KeyDown(object sender, KeyEventArgs e)
        {if (e.KeyCode == Keys.Enter) {txtDes.Focus();}}

        private void txtDes_KeyDown(object sender, KeyEventArgs e)
        {if (e.KeyCode == Keys.Enter){textDate.Focus();}

        private void textDate_KeyDown(object sender, KeyEventArgs e)
        {if (e.KeyCode == Keys.Enter) {txt_P_Price.Focus();}}

        private void txt_P_Price_KeyDown(object sender, KeyEventArgs e)
        {if (e.KeyCode == Keys.Enter) {txt_S_price.Focus();}}

        private void txt_S_price_KeyDown(object sender, KeyEventArgs e)
        {if (e.KeyCode == Keys.Enter) {txtqte.Focus();}}

        private void txtqte_KeyDown(object sender, KeyEventArgs e)
        {if (e.KeyCode == Keys.Enter) {cmbCategories.Focus();}}

        private void cmbCategories_KeyDown(object sender, KeyEventArgs e)
        {if (e.KeyCode == Keys.Enter) {btnAdd.Focus();}}

        private void button1_Click(object sender, EventArgs e)
        {
            PL.FORM_PURC_DATILS fm = new FORM_PURC_DATILS();
            fm.ShowDialog();
            if (fm.DGV_PUR_MANGER.CurrentRow.Cells[7].Value.ToString() != "Add")
            {
                txtRef.Text = fm.DGV_PUR_MANGER.CurrentRow.Cells[2].Value.ToString();
                txtDes.Text = fm.DGV_PUR_MANGER.CurrentRow.Cells[3].Value.ToString();
                txt_P_Price.Text = fm.DGV_PUR_MANGER.CurrentRow.Cells[4].Value.ToString();
                txtqte.Text = fm.DGV_PUR_MANGER.CurrentRow.Cells[5].Value.ToString();

                fm.Close();
            }
        }
    }
}
```

CATEGORIES Form

```
namespace smms
{
    public partial class FORM_CATEGORIES : Form
    {
        SqlConnection sqlcon = new SqlConnection("Data Source=DESKTOP-univerdity;Initial
                                                  Catalog=SMMS;Integrated Security=True");
        SqlDataAdapter da;
        DataTable dt=new DataTable();
        BindingManagerBase bmb;
        SqlCommandBuilder cmdb;
        public FORM_CATEGORIES()
        {
            InitializeComponent();

            textDescription.Enabled = false;
            btnEdit.Enabled = false;
            btnDelete.Enabled = false;
            PrintCurrentCateg.Enabled = false;
            ExportToExcelCurrent.Enabled = false;

            da = new SqlDataAdapter("select CatID as 'ID',CatName as 'Category' from Category",sqlcon);
            da.Fill(dt);
            DGV_List.DataSource = dt;
            textID.DataBindings.Add("text", dt, "ID");
            textDescription.DataBindings.Add("text", dt, "Category");
            bmb = this.BindingContext[dt];
            lbl_position.Text=(bmb.Position+1) + " / " + bmb.Count;
        }

        private void btnAdd_Click(object sender, EventArgs e)
        {
            if (textDescription.Text == string.Empty)
            {
                MessageBox.Show("write the name of category", "Attention", MessageBoxButtons.OK,
                                                                           MessageBoxIcon.Information);
                textDescription.Focus();
                return;

            }
            bmb.EndCurrentEdit();
            cmdb = new SqlCommandBuilder(da);
            da.Update(dt);
            MessageBox.Show("Add Successfully", "Add", MessageBoxButtons.OK,
                                                       MessageBoxIcon.Information);
            btnAdd.Enabled = false;
            btnNew.Enabled = true;
            lbl_position.Text = (bmb.Position + 1) + " / " + bmb.Count;
        }

        private void ExportToExcelCurrent_Click(object sender, EventArgs e)
        {
            RPT.RPT_SINGLE_CATEGORY myreport = new RPT.RPT_SINGLE_CATEGORY();
            ExportOptions export = new ExportOptions();
            DiskFileDestinationOptions dfdoptions = new DiskFileDestinationOptions();
            PdfFormatOptions pdfformat = new PdfFormatOptions();
            dfdoptions.DiskFileName = "E:\\CategoryDetails.pdf";
            export = myreport.ExportOptions;
            export.ExportDestinationType = ExportDestinationType.DiskFile;
            export.ExportFormatType = ExportFormatType.PortableDocFormat;
```

```
        export.ExportFormatOptions = pdfformat;

        export.ExportDestinationOptions = dfdoptions;
        myreport.SetParameterValue("@id", Convert.ToInt32(textID.Text));
        myreport.Export();
        MessageBox.Show("List Exported Successfuly ! ", "Export", MessageBoxButtons.OK,
                                                                  MessageBoxIcon.Information);
        btnAdd.Enabled = false;
        btnNew.Enabled = true;
        btnEdit.Enabled = false;
        btnDelete.Enabled = false;
        PrintCurrentCateg.Enabled = false;
        ExportToExcelCurrent.Enabled = false;
    }

    private void button12_Click(object sender, EventArgs e)
    {Close();}

    private void button_First_Click(object sender, EventArgs e)
    {bmb.Position = 0; lbl_position.Text = (bmb.Position + 1) + " / " + bmb.Count; }

    private void button_last_Click(object sender, EventArgs e)
    {bmb.Position = bmb.Count;lbl_position.Text = (bmb.Position + 1) + " / " + bmb.Count;}

    private void button_previous_Click(object sender, EventArgs e)
    {bmb.Position -= 1;_position.Text = (bmb.Position + 1) + " / " + bmb.Count; }

    private void button_Next_Click(object sender, EventArgs e)
    {bmb.Position += 1; lbl_position.Text = (bmb.Position + 1) + " / " + bmb.Count;}

    private void btnNew_Click(object sender, EventArgs e)
    {
        bmb.AddNew();
        btnNew.Enabled = false;
        btnAdd.Enabled = true;
        btnDelete.Enabled = false;
        btnEdit.Enabled = false;

        int id = Convert.ToInt32(dt.Rows[dt.Rows.Count ][0])+1;
        textID.Text = id.ToString();
        textDescription.Focus();
        textDescription.Enabled = true;

    }

    private void btnDelete_Click(object sender, EventArgs e)
    {
        bmb.RemoveAt(bmb.Position);
        bmb.EndCurrentEdit();
        cmdb = new SqlCommandBuilder(da);
        da.Update(dt);
        MessageBox.Show("Deleted Successfully", "Deleted", MessageBoxButtons.OK,
                                                           MessageBoxIcon.Information);
        lbl_position.Text = (bmb.Position + 1) + " / " + bmb.Count;
        btnAdd.Enabled = false;
        btnNew.Enabled = true;
        btnEdit.Enabled = false;
        btnDelete.Enabled = false;
        PrintCurrentCateg.Enabled = false;
        ExportToExcelCurrent.Enabled = false;
    }
```

```
private void btnEdit_Click(object sender, EventArgs e)
{
    bmb.EndCurrentEdit();
    cmdb = new SqlCommandBuilder(da);
    da.Update(dt);
    MessageBox.Show("Edited Successfully", "Edited", MessageBoxButtons.OK,
                                                     MessageBoxIcon.Information);
    lbl_position.Text = (bmb.Position + 1) + " / " + bmb.Count;
    btnAdd.Enabled = false;
    btnNew.Enabled = true;
    btnEdit.Enabled = false;
    btnDelete.Enabled = false;
    PrintCurrentCateg.Enabled = false;
    ExportToExcelCurrent.Enabled = false;
}

private void PrintCurrentCateg_Click(object sender, EventArgs e)
{
    RPT.RPT_SINGLE_CATEGORY rpt = new RPT.RPT_SINGLE_CATEGORY();
    RPT.FRM_RPT_PRODUCT frm = new RPT.FRM_RPT_PRODUCT();
    rpt.SetParameterValue("@id", Convert.ToInt32(textID.Text));
    frm.crystalReportViewer1.ReportSource = rpt;
    frm.ShowDialog();

    btnAdd.Enabled = false;
    btnNew.Enabled = true;
    btnEdit.Enabled = false;
    btnDelete.Enabled = false;
    PrintCurrentCateg.Enabled = false;
    ExportToExcelCurrent.Enabled = false;
}

private void PrintAllCateg_Click(object sender, EventArgs e)
{
    RPT.RPT_ALL_CATEGORIES rpt = new RPT.RPT_ALL_CATEGORIES();
    RPT.FRM_RPT_PRODUCT frm = new RPT.FRM_RPT_PRODUCT();
    rpt.Refresh();
    frm.crystalReportViewer1.ReportSource = rpt;
    frm.ShowDialog();
}

private void ExportToExcelAll_Click(object sender, EventArgs e)
{
    RPT.RPT_ALL_CATEGORIES myreport = new RPT.RPT_ALL_CATEGORIES();
    ExportOptions export = new ExportOptions();
    DiskFileDestinationOptions dfdoptions = new DiskFileDestinationOptions();
    PdfFormatOptions pdfformat = new PdfFormatOptions();
    dfdoptions.DiskFileName = "E:\\Categorieslist.pdf";
    export = myreport.ExportOptions;
    exprt.ExportDestinationType = ExportDestinationType.DiskFile;
    export.ExportFormatType = ExportFormatType.PortableDocFormat;
    export.ExportFormatOptions = pdfformat;
    export.ExportDestinationOptions = dfdoptions;
    myreport.Refresh();
    myreport.Export();
    MessageBox.Show("List Exported Successfuly ! ", "Export", MessageBoxButtons.OK,
                                                              MessageBoxIcon.Information);
}

private void DGV_List_MouseClick(object sender, MouseEventArgs e)
```

```
        {
            btnEdit.Enabled = true;
            btnDelete.Enabled = true;
            textDescription.Enabled = true;
            btnAdd.Enabled = false;
        }


 private void DGV_List_DoubleClick(object sender, EventArgs e)
        {
            btnAdd.Enabled = false;
            btnEdit.Enabled = true;
            btnDelete.Enabled = true;
            PrintCurrentCateg.Enabled = true;
            ExportToExcelCurrent.Enabled = true;
        }

        private void textID_KeyDown(object sender, KeyEventArgs e)
        {if (e.KeyCode == Keys.Enter) {textDescription.Focus();}}
}
}
```

ملخص المشروع باللغة العربية

إدارة السوبر ماركت المقترحة هو نظام تطبيقي قائم على نظام ال Windows ، ويهدف إلى أتمتة محلات السوبر ماركت المتوسطة والصغيرة من خلال جعل النظام موثوقًا وسريعًا وسهل الاستخدام وغني بالمعلومات. يقلل من الأعمال الورقية ومتطلبات القوى العاملة ويزيد من إنتاجية السوبر ماركت.

باستخدام هذا التطبيق ، يمكن للمرء إضافة التفاصيل وتعديلها وتحديثها وحفظها وحذفها وطباعتها. هناك أيضًا ميزة بحث للعثور على المنتجات المتوفرة في السوبر ماركت.