

COMPUTERIZATION OF STOCK AND SALES
OPERATIONS (A CASE STUDY OF ABUJA
CAPITAL MOTORS NIGERIA LIMITED)

BY

BIRAM AUGUSTINE
PGD/MCS/2004/2005/1172

A PROJECT SUBMITTED TO THE DEPARTMENT OF
MATHS/COMPUTER SCIENCE FEDERAL UNIVERSITY OF
TECHNOLOGY MINNA IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE AWARD OF THE
POSTGRADUATE DIPLOMA IN COMPUTER SCIENCE

January 2007

COMPUTERIZATION OF STOCK AND SALES
OPERATIONS (A CASE STUDY OF ABUJA CAPITAL
MOTORS NIGERIA LIMITED)

BY

BIRAM AUGUSTINE
PGD/MCS/2004/2005/1172

DEPARTMENT OF MATHS/COMPUTER SCIENCE FEDERAL
UNIVERSITY OF TECHNOLOGY MINNA

January 2007

APPROVAL PAGE

This project work has been read and approved by the undersigned, as meeting the requirements of the Department of Mathematics/Computer Science, Post Graduate School, Federal University of Technology, Minna.

Mr.M.Jiya
(Project Supervisor)

Date

Dr N.I. Akinwande (HOD)

Date

EXTERNAL EXAMINAR

Date

DEDICATION

This piece of work is dedicated to my beloved sisters and Chabiya Javan who stood by me.

ACKNOWLEDGEMENT

I wish to express my sincere and profound gratitude and appreciation to my project supervisor Mr.M.Jiya for his advice and suggestions of all during the course of this project. His influence and incentive on my writing ability and techniques are highly appreciated. Also my regards go to the HOD, Dr N. I. Akinwande, Prof K.R. Adeboye, Dr L. N. Ezeaku, Dr Y. M. Aiyesimi, M.Tech Coordinators, Dr Abubakar U.Y, the Post Graduate Diploma coordinator Mr.Ndanusa Abdulrahaman, and to all lecturers in the Department.

Also I am highly indebted to the following people Musa Audu, Miss Chabiya Javan ,Mrs. Josephine Haske, Mrs Regina Clifford, and to my collogues Reuben James, Barr Habiba R.Alhassan, Margaret Loko and the Management of FHA Homes Limited.

ABSTRACT

This Project work is aimed at computerization of stock and sales of Abuja Capital Motors for easy access of business documents.

CHAPTER ONE

1.0

INTRODUCTION

Both the private and government sector nowadays lay emphasis on strategic planning balancing of records control and rapid response to development problem. The result is a constant demand for better and more timely information. The need for efficient planning and controlling cannot be over emphasized. This implies to a good system of management. This is especially true in the area keeping accurate records of client (customers) of an organization to minimize fraud misplacement of records of customers, and errors in terms of calculation etc.

For the effect of sound management practice in customer relations development which help with the up to date and accurate customer information on which decision making by the organization or management body are based.

The need for accurate customer records of clients is high nowadays due to increase inefficiency, productivity, response time to customers, good relationship with customers, proper handling of customer records etc.

From the management view point, the customer relations department exists for two reasons, first they provide information to the management, thus making available customer data about each client such as customer number, customer name, customer city, customer state etc.

Secondly the customer relation department acts as a controlling agency. In this regard, it is responsible for the welfare of the customers in terms of monitoring the customers altitude towards the goods and services rendered or given to them by the company and to check whether they are being satisfied or not .As the popular saying goes “if we satisfy

you tell others, if not tell us". And this remark is due to the fact or slogan that says, "Customers are always right." Abuja capital motors limited is an organization that deals with the sale and services of different brands of motor vehicles.

When a motor vehicle is purchased or bought from the organization by an individual, a record or file is said to be opened for that individual on long term business relationship between the individual and the organization. And at this time the individual becomes a customer.

The long term relationship of the organization and the individual is said to have its benefits and such benefits as follows.

1. 1 to 2years guarantee on the vehicle i.e. giving limited assurance of the durability of the vehicle and component part.
2. Free of charge services of the customers vehicle for about 1-3years i.e. this is an attractive offer that the management of the organization intends to use in order to keep its customer.
3. Sales and serving of the customer's vehicles at discount rate.
4. The organization can bring his /her vehicle any for repair(s)
5. The organization can pick up the customers cars for repairs and take it back after the work has been properly done and the customer can pay up later.
6. The customer will have a credit facility after some few months or years.

The organization incorporates other types of facilities such as

1. Importing a customer's side vehicle for a customer or individuals that does not have a record or file with the organization i.e. shipping or cargoing a vehicle from one country to another or state to state.

2. Customerization of customer's vehicles by the organization. A customer can request for some additional car accessories to his /her vehicle and such accessories include - Alloy wheels, lather seats, compact disc players, Tinted screen (windows) extra shocks absorbers e.t.c
3. The organization can also assist the customer with him /her insurance. If he / she insist or request for it to be done by the organization.
4. The organization can also help a customer in selecting or choosing of a vehicle that will best suit his / her state and environmental conditions.

1.1 PROBLEM IDENTIFICATION

The Abuja Capital Motors Limited is an organization that deals with the sales and services of different brands of motor vehicles. All products available in stock and sales of products are recorded manually with the use of paper file.

On study the existing system, the following problems were observed.

- High rates of errors:-Due to cumbersome nature of the system using files, there is high rate of making mistake, which leads to low rate of transaction and inefficient distribution of stock products.
- Misplacement of files:-As the population of customers increases, some of the files are misplaced after accessing it for customer information. This can leads to several days of delay in processing and products to customers.
- Difficult in report generation:-Statistical and Management information reports are always tedious to realize due to the large volume of records involved at all time.

- Inadequate Security and Privacy of files:-files are not properly kept, which resulted into fraudulent manipulation of data, loss of data processing facilities, disclosure of sensitive data, loss of data as a result of error.
- Time factors:-A lot of time is wasted on answering customer, which can leads to shift of customer from the Company or Organization.
- Efficiency:-Correction of errors is tedious through this system.
- Large space involved:-Since the

1.2 THE OBJECTIVES OF THE PROJECT

The general aim of the project is to investigate the customer information system Abuja capital motors Nigeria Limited is sufficient depth to be able to provide information that justifies the development of a new system and also identify the basic information requirements.

The specifies objectives are:

1. To supply the availability of up to date management information system to monitor customer records time to customer, good relationship with customer, customers accounts e.t.c
2. To smoothen the flow of such document from appropriate locations to the computer center.
3. To revise set of source documents so as to extract the required information for input data.
4. To study the control of customer records through documentation and computer system with an integration with customer relations department.

5. To study the training of staff on the new system.
6. To make recommendation based on the above findings.

1.3 SCOPE OF THE PROJECT

It is quite obvious that the coverage of project work of the nature should be limited with the subject matter, hence it would have been of a more expository, and exhaustive work if all business organization was reached. Due to the fact that various factors were encountered therefore much emphasis has been expedited to nevertheless survey and present as a classical review of customer information system: A case study of Abuja Capital Motors Nigeria limited

This case study is a representative sample of a related business organization in the state as application of the concept procedure and techniques of customer information system which include customer data only.

There are some constraints associated with computerization of customers information system which include the steps that will be taken to ensure that the new system is implemented wisely and carefully so as to be able to general appropriate revenue in returns.

CHAPTER TWO (LITERATURE REVIEW)

2.1 THE OLD SYSTEM

The old or current system is said to be the manual data collection method.

MANUAL

This means that data relating to business transaction are received by clerks in the forms of source documents. Processing is done mentally with aid of calculations or an adding / listing machine.

Storage is done using ledger cards and storage media is usually cabinets.

The outputs are usually statement of accounts and ledger books / cards and are manually prepared.

The advantages of manual processing are intangible but the disadvantages are

- i. Processing speed is slow.
- ii. The system or method is not reliable.
- iii. Lack of accuracy in terms of calculation.
- iv. The method of data collection is not flexible.
- v. Storage facility is not safe. Anybody can temper with the cabinet and change customer records.
- vi. The cabinets require too much floor space in the organization.

Finally, the in proper keeping of customer records about a customer makes it cumbersome or could lead to misplacement of vital information about a customer records.

2.2 THE NEW SYSTEM

The proposed new system is the computerization of the old or current system. The need to computerize the old system was due to the fact that it had many disadvantages, which was disorganizing the transaction of the company. These brought about the introduction of the electronic data processing.

2.3 WHAT IS ELECTRONIC DATA PROCESSING?

Electronic data processing involves the use of a computer system. A computer system is an electronic machine that is capable of performing difficult tasks by executing a set of instructions which are held within it as a set of numbers.

The computer is capable of

- Taking in an input data and set of instruction for processing such data.
- Processing such data by following the set of instructions and
- Processing results or outputs.

Computers are widely used today because of the following advantages which they have over most of the other known method of processing data. The advantages are

- i. High processing speed.
- ii. Large storage space.
- iii. Flexibility.
- iv. Reliability.
- v. Ability to perform repetitive task
- vi. Affordability

vii. Maintenance.

viii. Portability.

When computer system is introduced into an organization, in most cases it is to support the work already been or enhancement. Although the working of the future system may defer substantially from those of the current system. The information held and the major functions often remain relatively future system. The objectives of the requirements analysis are as follows.

- i. To determine the scope of the application.
- ii. To establish how to integrate information technology components with other needs.
- iii. To form a view of overall system costs and benefits.
- iv. To confirm the viability of continuing further.

The requirements analysis is driven by requirement definition and business system options. This places the study in the contents of the future system..

In order to enable to design and justify the new system, several aspects of the system are investigated.

1. OPERATION ON DATE.

In understanding the current operations and data requirements are more easily understood.

2. PROBLEMS INHERENT IN THE CURRENT SYSTEM.

Examining the problems that are encountered in the current system and to ensure that they are not replicated in the new system.

3. BOUNDARIES OF THE SYSTEM

It is important throughout the project the analyst should attempt to draw accurate boundaries around the area of investigation to avoid needless efforts on areas that lie outside the boundaries or more importantly to ensure that all of the areas inside the boundaries are investigated. The boundaries of the current system investigation should be explicitly stated and agreed with those concerns.

4. COST AND VOLUMES

This acts as a base for system justification and sizing in the analysis of the requirements.

2.4 COMPARISONS OF THE OLD SYSTEM AND NEW SYSTEM.

The comparisons of the old and new systems is such that the old or current system is said to be lacking the following

- **PROCESSING SPEED IS SLOW**

In terms of response time to customer is very discouraging.

- **THE METHOD OF DATA COLLECTION IS NOT RELIABLE**

Because data cannot be easily modified or corrected without it making the data entry book rough.

- **THE STORAGE FACILITY IS NOT SAFE**

Because anybody can enter the cabinet and change some vital customer information.

On the other hand, the introduction of the computer system, the new proposed system can improve the efficiency of workers in the company and will help the effective decision making in the management. The new proposed system will help the company in the following way

1. HIGH PROCESSING SPEED

In terms of response time to customer will be fast and brief.

2. PORTABLE

In terms of floor space required by the system is not much.

3. FLEXIBILITY

The same computer system can be used for various job or application. All you need to do is to change the set of instruction.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

3.0 INTRODUCTION TO SYSTEM ANALYSIS

This chapter simply discussed the relative study undergone in carry out this project. It also emphasized the method of collecting information or needed raw fact to be processed. This section projecting the existing method i.e. the current method in used to be upgraded or changed .It briefs on the problems associated with the current problems and finally reports the analogy of the proposed system.

3.1 STATEMENT OF THE PROBLEM

This is the point at which the proposed new system gains its first exposure or securing before a new design and development effort takes place. A problem is identify, a typical motivation for study a system is problem experience by the user of the system.

As another possibility, the customer Relations Department may be motivated perhaps by a change in technology to call new opportunity to do thing in a better and more efficient way. The fact is that the potential problem due to failure to capitalize new techniques can carry cost penalty.

Still another motivation for initiating an investigation can come from external forces such as the government, rivals (other companies), customer etc and its regulation causes many system design changes e.g.

MANUAL → AUTOMATED → COMPUTERISED SYSTEM

The main purpose of the investigation is to determine

- If these is a problem.

- The extent or seriousness of the problem.
- Whether it is processed with a full scale study of the problem and the alternative solution. Therefore, project accomplishment of the definition phase includes details of the above activities.

3.2 SYSTEM ANALYSIS

System analysis is the study and analysis of problems of an existing information system and identification of the requirements for the solution. The information gathered during the feasibility study is the jotting of parts for the detailed system analysis. System analysis on the other hand, is the activities concern with collecting and analysis of facts about an existing system as a preliminary to designing computerized system. A preliminary investigation is normally undertaken.

Personal observation of current procedures during this fact gathering phase of the system study, the analyst must develop a critical and questioning approach to each of the procedures with the system to determine what is actually taking place. For too often in many organizations, operations are being performed not because they are efficacious or effective, but because they have always been done this way.

Specific steps are undertaken at this time to gather some vital information.

- Review of the organizational chart of the structure of the organization to determine who the workers are, supervisors Department, which is associated with the system under study.
- Conduct interviews with selected personnel in customer Relations Department to determine what is actually taking place in the current

system, rather than what is supposed to take place according to written rule of the organization.

- Obtain actual copies of operating document, all the data flow with the current existing system and actual procedures which are followed.

Prior to analyzing the facts to determine why certain procedures are followed and what can be done to improve them, the fact as found must be documented. Although a number of methods exist on how to document what takes place within a system, one of the more effective is the data flow diagram as contained in this chapter.

The data flow diagram graphically illustrates the flow of data and the procedures that is used for a customer record system by detailing the processing occurring in the present system in that manner, the research can examine each of the steps involved to determine exactly what happens in the present system.

Once the system has been documented, the analyst must review it. This analysis is performed to detect any flow or errors which are contained within the system. Since the present system is being reviewed in detail, it is quite obvious that there are some problems in the system. The purpose of this analysis stage is to find those problems and develop some possible solutions.

The present system must be criticized against principle of procedures after which the strengths and weakness of the system should be apparent.

Once the detailed investigation and analysis have been completed, a presentation will normally be given to management recommending one or more approaches to solving the problem. This presentation will normally include the definition of the problem which has

been uncovered, recommendations as to possible solutions and the costs associated with possible solutions.

3.3 SYSTEM ANALYSIS AND DESIGN THE MANUAL SYSTEM.

The Abuja Capital Motors Nigeria limited currently uses the manual system in keeping stocks and sales of items. The Abuja Capital Motors operations are manned by two level of people the sales clerk and the manager. The sales clerk handles sales while the manager handles an updating of stocks, changing of prices etc. The inputs come from sale entered by the clerk and store voucher forms, while the output from the system is receipts and periodical sales account preparation sheets.

As expected therefore the manual system involves overtime, the use of large volumes of papers to be kept and moved when the need arises. When information is required the papers have to be studied.

Obviously this made some inaccuracies which exist in the present manual system inevitable, stock taking and rendering of account at the end of the week or at the end of the month usually poses a headache because they are been done manually . The manual system is known to have the following advantages.

- (a) There is redundancy of data.
- (b) Information is not accurate.
- (c) Integrity and security of data cannot be vouched for.
- (d) There is delay in terminal report preparation.
- (e) It gives room from several undetected case of fraud.

However, it is highly pertinent to point out that the manual system is cost effective and cheap in the short run, it also provides for quick error handling procedures.

TIME

In view of the time constant, it is vital to consider time taken in development the new system in training of the user department (staff of the organization) associated with the system study in order to be able to be familiar with the new system to enhance productivity and efficiency in the day-to-day smooth running of the system (that is when response time to customer as a key factor is very important)

RECRUITMENT

This service of an expert may be required in handling or controlling new system. The last is the financial constraint, it is centered on cost. Certainly the proposed project will no doubt be expensive because of the cost of acquiring new software and computer system. Recruiting efforts are expensive. When an organization considers various recruiting sources it does so with some sense of effectiveness in mind that is, recruiting expenditures are made where the best return on the investment can be realized. Unfortunately, because of limited resources the expenditure needs to be place on priority.

3.4 PROGRAM DEVELOPMENT ENQUIRY

This section of the project will enable the system analyst to give preliminary investigation report to the management personel. The fact gathering of the investigation may be in two parts.

1. Personal interviews with management personnel.

2. Questioners with the customer relation, department or any body in the organization that deals directly with the customers.

The reason for the personal interview with the management personnel is to establish whether there is the need for a new system and if so to specify the objectives of the system. Another basic fact to which interviews and questionnaires were used was to uncover the true nature of the problems and to also gather the current system documentation and operating forms of the system.

Personal observation of current problems during this fact gathering phase of the system study, the analyst must develop critical and questioning approach to each of the procedure within the system to determine what is actually taking place. For too often in both private and government sectors, operations are being performed not because they are efficacious or effective, but because they have always been done this way.

PROGRAMMING LANGUAGE USED

The programming software that was chosen for this project is the C#.net. it is also functions and can be used as a database management programme.

WHAT IS DATA BASE

A database is a collection of information organized and presented to serve a special purpose.

COMPUTER DATABASE

A computer database is an organized collection of information stored on a computer disk and accessed with a database management system.

WHAT IS THE DATABASE MANAGEMENT SYSTEM

A database management system (DBMS) is a computer program which allows the user to manipulate the database.

3.5 FUNCTIONS OF A DATABASE MANAGEMENT SYSTEM

The purpose of a database management system is to organized data into easy to use information to achieve this goal; most database management software programs offer the following features.

1. DATABASE DEFINITION

This feature enables the user to create the structure of the database. The fields for each record are Identified and defined by naming each field's length and by establish the type of data to be stored in each field.

2. DATA ENTRY

This feature enables the user to enter information into database.

3. DATA MODIFICATION

This feature enables the user to correct or modify data in the database. If the information changes or was wrongly entered, it can be altered using the modification capabilities of the database management program.

4. DATA SEARCHING

This feature enables the user to look through the database to find specific information. This capability could be used to get the names and addressed to all of company's customer living in particular state or city.

5. DATA INDEXING

An index is used to keep the information in a database in alphabetical or numerical order automatically.

6. DATA SORTING

This feature enables the user to arrange non index data alphabetical or numerical sequence. Using this capability usually makes the information easier to understand.

7. REPORT GENERATION

This feature enables the user to print the data in a useful manner.

8. INFORMATION STORAGE AND RETRIEVAL

It handles storage, retrieval and organization of information in efficient manner better than other micro – computer base DBMS.

9. PROGRAM COMMAND

Its commands are short and easy to coordinate and remember.

10. DEBUGGING

It also includes some features of tracking unwanted program bugs. Few programming languages after such facilities.

3.6 DESIGN OF THE NEW SYSTEM

The system is presented in modular form with the main menu consisting of options programs on modules of cars, print, add user, and exit which are well structured to enhance times and efficiency.

3.7 SOURCE PROGRAM DOCUMENTATION

The modules are referenced by the main system menu. To open these programs double click the Car Shop, User name and password, and this takes you to the main system Menu.

DISCRIPTION OF PROGRAM MODULES

The main modules are divided into four sub modules namely:

CAR

The sub module enable the user to view all cars in stock, also it allow you to add to the stock. This system enables the user to log into sales and also to check sales history of the company.

PRINT

This module enables the user to produce reports on list of cars in stock and sales log

ADD USER

The routine enable you to add many users in the company.

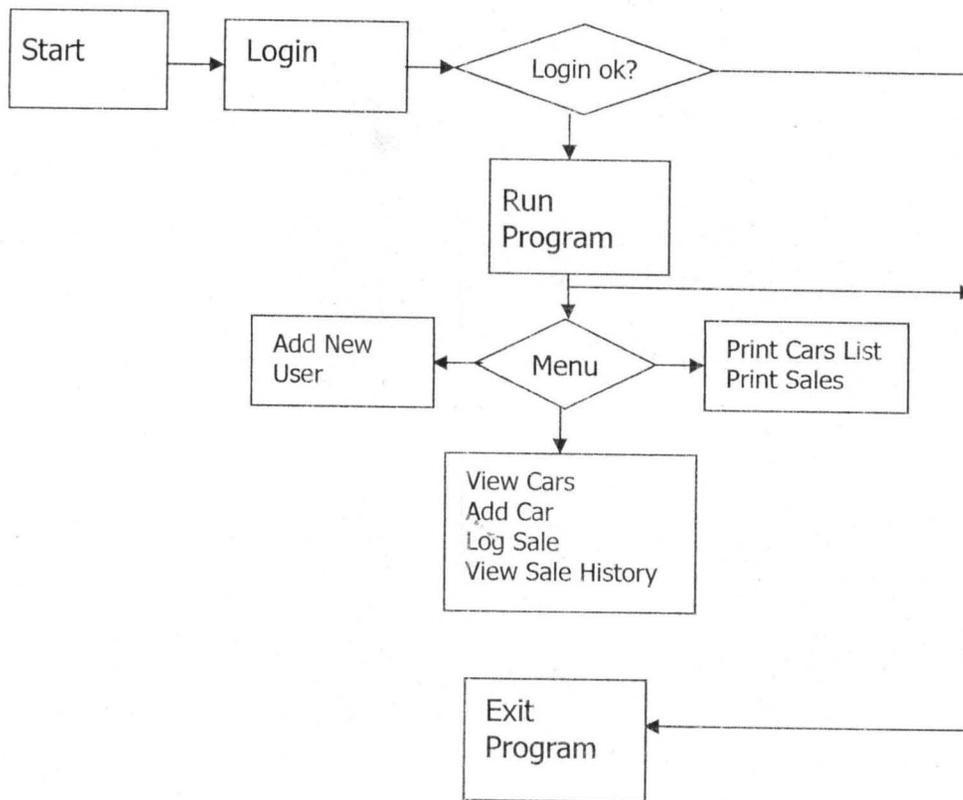
EXIT

When all data have been keyed into the system and all processes completed, next is to quite the record, which takes you back to the normal environment.

3.8 LIMITATIONS AND CAPABILITIES OF THE PROGRAM

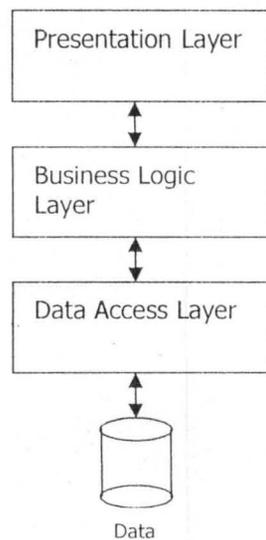
The program works efficiently if correct data are keyed in. Wrong input will generate wrong output.

The program has capability of storing very large record; this is made possible by use of DBMS, which help in managing the database files.



Source Code for Car Shop

The Application is divided into 3 layers; The **Data Access Layer** which deals with the retrieval and storage of data, The **Business Logic Layer**, which deals with the objects and rules of the business, **The Presentation Layer**, which deals with the user interface and presentation of the data.



CHAPTER FOUR

IMPLEMENTATION

4.0 SYSTEMS IMPLEMENTATION

Successful systems after relative period of development are implemented and subsequently operated by users who obtain support and guidance from the systems department as required. The task of management is then to see to a smooth transition of the systems into operational state and thereafter maximized the benefits from the implemented system by working with the users to monitor and adjust its performance. Implementation refers to the transition of a system from construction stage to live stage.

4.1 WORK STATION REQUIREMENT

This refers to computer hardware and software required for smooth take –off of the computerization exercise, which is as follow

4.1.1 HARDWARE REQUIREMENT

To meet the desired objectives, that is, the computerization of stock and sales operations. It is proposed that it be run on:

Pentium III, or IV systems processor with 128MB RAM on Mother with 2.36HDD, the systems should also have CD-ROM drive and 3.5 drive.

4.1.2 SOFTWARE REQUIREMENT

The system should be run on window operating system, window 2000 and window XP

4.2 SYSTEM TESTING

Before implementation and on acquired hardware and other peripherals, the program will be loaded and test run for 4 weeks to ascertain its efficiency and where necessary, make modifications.

4.3 SYSTEM CHANGE OVER

File conversion is a vital activity, which is sometime under estimated. The conversion simply put, is the process of converting the old file data into the form required by the new system. It is often referred to as part of change over.

Infact, file conversion is often a complete and separate system task in itself involving fact finding, analysis, design of manual methods and computer processing among others. The change over from old to the new system may take place when:

- The system has been proven to the satisfaction of the system analysis and other implementation activities have been completed.
- User managers are satisfied with the results of the system test, staff training and reference manuals.
- The target date for change over is due.

The change over may be achieved in numbers of ways. The most common methods are.

- Direct change over
- Parallel change over
- Pilot change over
- Stage change over

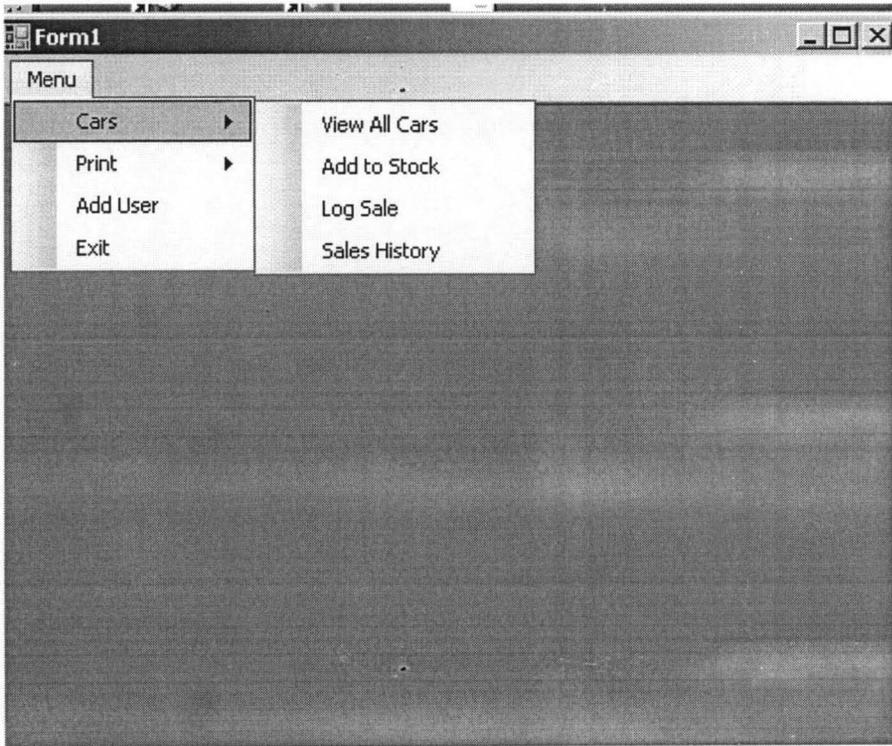
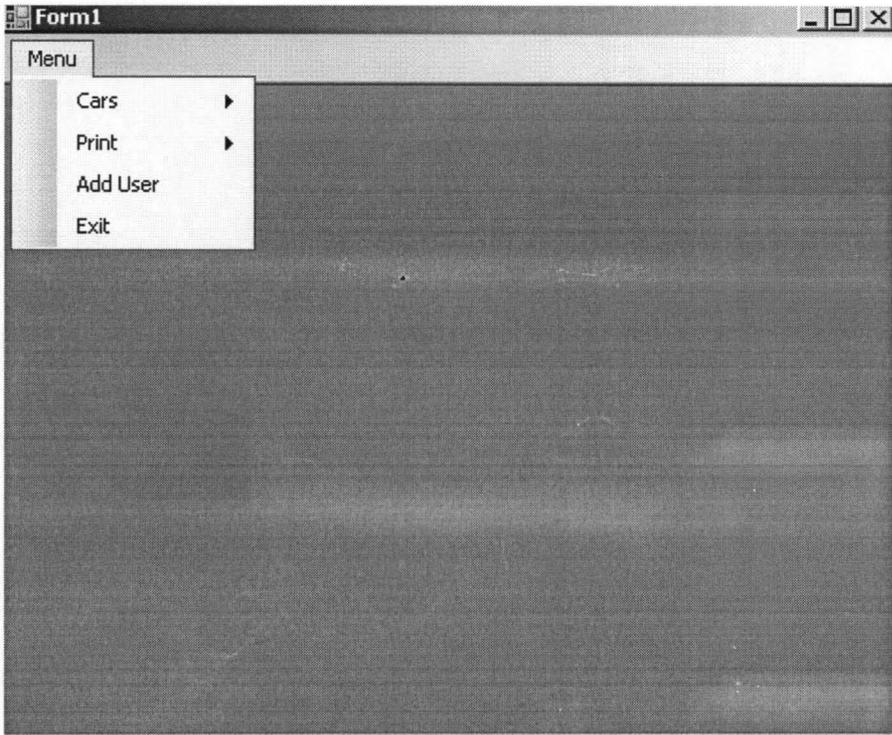
Direct change over: This method is the bold move which should be undertaken when everyone involved has confidences in the new system. When a direct change over is planned, system test and training should be comprehensive and the change over itself planned in detail. This method is potentially the least expensive but the most risky.

Parallel change over: This means processing current data by the old and new system side by side so as to cross-check the result. Its main attraction is that old system is kept alive and cooperation with the new system until the new system has been proven satisfaction beyond all reasonable doubt. This system allows the result of the new system to be compared with the old system before acceptance by the user thereby promoting user confidence. The only demerit of this method is the attendance cost of running two systems at the same time.

Pilot Running: This is similar in concept to parallel running. Data from one or more previous period, for the whole or part of the system is run on the new system after results are compared with the old .It is not as destructive as parallel operation, since timing is less critical.This method is more like an extended system test but it may be considered a more practical form of change over for organizational reason.

Stage Changeover: This involves a series of limited size direct changeovers, the new system being introduced piece by piece. A complete part or logical section is committed to the new system while the remaining parts or sections are processed by the old system. Only when the selected part is operating satisfactorily is the remainder transferred. This method reduces the risk

inherent in a direct change over of the whole systems and enables users to learn from mistakes made as the changeover progresses. The demerits of this method include problems of controlling the selected parts of the old and new system and it tends to prolong the implementation period.



Form1

Menu

frmAllCars

All Cars | Add Car | Log Sale | Sales History

Manufacturer

Model

Total

Add Car

Form1

Menu

frmAllCars

All Cars | Add Car | Log Sale | Sales History

Manufacturer

Model

Total

Log Sale

List of All Cars

Manufacturer	Model	Stock
Toyota	Camry	5
Honda	Civic	4
Toyota	Van	2
Toyota	Hiace	10
BMW	5 Series	8
Toyota	Jeep	10
BMW	X5	6
Mercedes	Benz	11
Kia	Optima	5

CHAPTER FIVE

5.1 SUMMARY OF FINDINGS

Since the advantages of a computerized marketing system of any given institution or business organization cannot be over emphasized, it is in this direction of information manipulation, access to data, duration means of storage system and easy retrieval of business documents and customer information records that the initiative for the computerization of the customer sales information of Abuja Capital Motors was conceived.

The research have no doubt that if a proper and effective computerized customer sales information system should be put in place, there will be an improved keeping of customers information , processing and early access to business records.

5.2 RECOMMENDATION

To ensure that the program achieve the set objectives, the following recommendation are made.

5.2.1 STAFF TRAINING

It is important that training of personnel should start off as soon as implementation is completed. All personnel detailed to on this system, therefore, will be required to under go training on how to work with the newly installed system.

In this process, users' guide should be provided to aid in effective training of staff. There should be operators. A resident programmer may be needed which will ensure that the system runs within acceptable standard. He will be in charge of software maintenance.

5.2.2 MAINTENANCE

As the new system is used, there might be need for added improvement. These improvements are made to prolong the system life cycle.

In this process of maintenance, all system documentation and user documentation must be referred to maintenance also could be either hardware or software. This is because as computer continues to advance with increase technology each day, and newest concepts.

This could be achieved by upgrading the memory to higher processing speed or by modifying the program so as to meet users' requirements. Although, there is no specific number of records the system can handle at any particular time, but as the need arises, any specific dimension can be introduced by a programmer as a way of effecting some minor changes.

The Presentation Layer

frmMain.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

using CarShop.BusinessFacade;

namespace CarShop
{
    public partial class frmMain : Form
    {
        public frmMain()
        {
            InitializeComponent();
        }

        private static User user;

        public static User User
        {
            get { return frmMain.user; }
            set { frmMain.user = value; }
        }

        private Form IsMdiChildOpen(Type typ)
        {
            foreach (Form frm in this.MdiChildren)
                if (frm.GetType() == typ)
                {
```

```

    {
        this.OpenCarsForm(CarFormTabs.AddCar);
    }

    private void logSaleToolStripMenuItem_Click(object
sender, EventArgs e)
    {
        this.OpenCarsForm(CarFormTabs.LogSale);
    }

    private void
salesHistoryToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        this.OpenCarsForm(CarFormTabs.SalesHistory);
    }

    private void carsToolStripMenuItem1_Click(object
sender, EventArgs e)
    {
        if (this.IsMdiChildOpen(typeof(frmAllCarsReport))
== null)
        {
            frmAllCarsReport frm = new frmAllCarsReport();
            frm.MdiParent = this;
            frm.Show();
        }
    }

    private void salesLogToolStripMenuItem_Click(object
sender, EventArgs e)
    {
        if (this.IsMdiChildOpen(typeof(frmSalesReport)) ==
null)

```

```

        {
            frmSalesReport frm = new frmSalesReport();
            frm.MdiParent = this;
            frm.Show();
        }
    }

    private void exitToolStripMenuItem_Click(object
sender, EventArgs e)
    {
        Application.Exit();
    }

    private void addUserToolStripMenuItem_Click(object
sender, EventArgs e)
    {
        if (this.IsMdiChildOpen(typeof(frmUser)) == null)
        {
            frmUser frm = new frmUser();
            frm.MdiParent = this;
            frm.Show();
        }
    }
}
}

```

frmAllCarsReport.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;

```

```
        frm.WindowState =  
FormWindowState.Normal;  
        frm.BringToFront();  
        return frm;  
    }  
    return null;  
}
```

```
    private void  
viewAllCarsToolStripMenuItem_Click(object sender,  
EventArgs e)  
    {  
        this.OpenCarsForm(CarFormTabs.AllCars);  
    }
```

```
    private void OpenCarsForm(CarFormTabs tab)  
    {  
        frmAllCars frmC =  
(frmAllCars)this.IsMdiChildOpen(typeof(frmAllCars));  
        if (frmC == null)  
        {  
            frmAllCars frm = new frmAllCars();  
            frm.MdiParent = this;  
            frm.CurrentTab = tab;  
            frm.Show();  
        }  
        else  
        {  
            frmC.CurrentTab = tab;  
        }  
    }
```

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

```

using System.Windows.Forms;

using CarShop.DataAccess;

namespace CarShop
{
    public partial class frmAllCarsReport : Form
    {
        public frmAllCarsReport()
        {
            InitializeComponent();
        }

        private void frmAllCarsReport_Load(object sender,
        EventArgs e)
        {
            using (CarsGateway cGateway = new
            CarsGateway())
            {
                // cGateway.GetAllCars(this.CarBindingSource);
                this.CarBindingSource.DataSource =
            cGateway.GetAllCars();
                this.reportViewer1.RefreshReport();
            }
        }
    }
}

```

frmLogin.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;

```

```

using System.Windows.Forms;

using CarShop.BusinessFacade;
using CarShop.DataAccess;

namespace CarShop
{
    public partial class frmLogin : Form
    {
        public frmLogin()
        {
            InitializeComponent();
        }

        private void Login_Load(object sender, EventArgs e)
        {
        }

        private void btnOk_Click(object sender, EventArgs e)
        {
            try
            {
                frmMain.User = new User(this.txtUserName.Text,
                this.txtPassword.Text);
                using (UsersGateway uGateway = new
                UsersGateway())
                {
                    uGateway.ValidateUser(frmMain.User);
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }
    }
}

```

```

    }
    finally
    {
        if (frmMain.User.Validated)
            this.Close();
        else
        {
            MessageBox.Show("Invalid Login");
            this.txtPassword.Clear();
            this.txtPassword.Focus();
        }
    }
}

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

```

frmSalesReport.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

using CarShop.DataAccess;

```

```
namespace CarShop
{
    public partial class frmSalesReport : Form
    {
        public frmSalesReport()
        {
            InitializeComponent();
        }

        private void frmSalesReport_Load(object sender,
EventArgs e)
        {
            this.reportViewer1.RefreshReport();
        }

        private void btnSearch_Click(object sender, EventArgs
e)
        {
            using (CarsGateway cGateway = new
CarsGateway())
            {
                if (this.cbGetAll.Checked)
                    this.SaleLogBindingSource.DataSource =
cGateway.GetAllSales();
                else
                    this.SaleLogBindingSource.DataSource =
cGateway.GetSalesWithinDate(this.dtpFrom.Value,
this.dtpTo.Value);
                this.reportViewer1.RefreshReport();
            }
        }
    }
}
```

frmUser.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

using CarShop.DataAccess;

namespace CarShop
{
    public partial class frmUser : Form
    {
        public frmUser()
        {
            InitializeComponent();
        }

        private void btnCreate_Click(object sender, EventArgs
e)
        {
            if (this.txtUserName.Text.Length == 0)
            {
                MessageBox.Show("Enter user name");
            }
            else if (this.txtPassword.Text.Length == 0)
            {
                MessageBox.Show("Enter password");
            }
            else
            {

```



```
    frmLogin lg = new frmLogin();
    lg.ShowDialog();
    if (frmMain.User.Validated)
        Application.Run(new frmMain());
    }
}
}
```

UIManager.cs

```
using System;
using System.Collections;
using System.Text;
using System.Windows.Forms;

using CarShop.BusinessFacade;

namespace CarShop.Services
{
    public class UIManager
    {

        public const string Manufacturer = "Manufacturer";
        public const string Model = "Model";

        public static void PopulateManufacturers(IEnumerable
source, ComboBox combo, string property)
        {
            combo.Text = string.Empty;
            combo.Items.Clear();
            foreach (Car car in source)
            {
                if (property == UIManager.Manufacturer)
                {
                    if (!combo.Items.Contains(car.Manufacturer))
                        combo.Items.Add(car.Manufacturer);
                }
            }
        }
    }
}
```

```

    }
    else if (property == UIManager.Model)
    {
        if (!combo.Items.Contains(car.Model))
            combo.Items.Add(car.Model);
    }
    }
    }
    }
}

```

The Business Logic Layer

CarFormTabs.cs

```

using System;
using System.Collections.Generic;
using System.Text;

```

```

namespace CarShop
{
    public enum CarFormTabs
    {
        AllCars = 0,
        AddCar,
        LogSale,
        SalesHistory
    }
}

```

Car.cs

```

using System;
using System.Collections.Generic;
using System.Text;

```

```
namespace CarShop.BusinessFacade
```

```
{
```

```
    public class Car
```

```
    {
```

```
        string make;
```

```
        string model;
```

```
        int stock;
```

```
        int id;
```

```
        public Car() { }
```

```
        public string Manufacturer
```

```
        {
```

```
            get { return this.make; }
```

```
            set { this.make = value; }
```

```
        }
```

```
        public int Id
```

```
        {
```

```
            get { return this.id; }
```

```
            set { this.id = value; }
```

```
        }
```

```
        public string Model
```

```
        {
```

```
            get { return this.model; }
```

```
            set { this.model = value; }
```

```
        }
```

```
        public int Stock
```

```
        {
```

```
            get { return this.stock; }
```

```
            set { this.stock = value; }
```

```
        }
```

```
}  
}
```

SaleLog.cs

```
using System;  
using System.Collections;  
using System.Text;  
  
namespace CarShop.BusinessFacade  
{  
    public class SaleLog  
    {  
        int id;  
        string user;  
        string make;  
        string model;  
        int amount;  
        DateTime date;  
  
        public SaleLog() { }  
  
        public int Id  
        {  
            get { return this.id; }  
            set { this.id = value; }  
        }  
  
        public string User  
        {  
            get { return this.user; }  
            set { this.user = value; }  
        }  
  
        public string Manufacturer  
        {
```

```
    get { return this.make; }  
    set { this.make = value; }  
}
```

```
public string Model  
{  
    get { return this.model; }  
    set { this.model = value; }  
}
```

```
public int Amount  
{  
    get { return this.amount; }  
    set { this.amount = value; }  
}
```

```
public DateTime Date  
{  
    set { this.date = value; }  
}
```

```
public string DateString  
{  
    get  
    {  
        return this.date.ToString("dd/MM/yyyy");  
    }  
}
```

```
    }  
}
```

User.cs

```
        get { return this.validated; }
        set { this.validated = value; }
    }
}
```

Data Access Layer

GatewayBase.cs

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Data.OleDb;

namespace CarShop.DataAccess
{
    public class GatewayBase : IDisposable
    {
        private OleDbConnection conn;

        public GatewayBase()
        {
            this.conn = new
OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=data.sip");
        }

        protected void ExecuteNonQuery(OleDbCommand
cmd)
        {
            try
            {
                if(this.conn.State !=
System.Data.ConnectionState.Open)
                    this.conn.Open();
            }
        }
    }
}
```

```
        cmd.ExecuteNonQuery();
    }
    finally
    {
        this.conn.Close();
    }
}
```

```
protected OleDbCommand GetCommand(string
commandText)
{
    OleDbCommand cmd = new
OleDbCommand(commandText, this.conn);
    return cmd;
}
```

```
protected OleDbDataReader
GetReader(OleDbCommand cmd)
{
    if (this.conn.State !=
System.Data.ConnectionState.Open)
        this.conn.Open();
    OleDbDataReader reader = cmd.ExecuteReader();
    return reader;
}
```

```
protected object ExecuteScalar(OleDbCommand cmd)
{
    object value = null;
    try
    {
        if (this.conn.State !=
System.Data.ConnectionState.Open)
            this.conn.Open();
        value = cmd.ExecuteScalar();
    }
}
```

```
    }  
    finally  
    {  
        this.conn.Close();  
    }  
    return value;  
}
```

```
protected void CloseConnection()  
{  
    this.conn.Close();  
}
```

```
protected void  
ExecuteNonQueryWithTransaction(OleDbCommand[] cmds)  
{  
    OleDbTransaction trans = null;  
    try  
    {  
        if (this.conn.State !=  
System.Data.ConnectionState.Open)  
            this.conn.Open();  
        trans = this.conn.BeginTransaction();  
        foreach (OleDbCommand cmd in cmds)  
        {  
            cmd.Transaction = trans;  
            cmd.ExecuteNonQuery();  
        }  
        trans.Commit();  
    }  
    catch  
    {  
        trans.Rollback();  
        throw;  
    }  
}
```

```
}

#region IDisposable Members

public void Dispose()
{
    this.conn.Dispose();
}

#endregion
}
}
```

CarsGateway.cs

```
using System;
using System.Collections;
using System.Text;
using System.Data.OleDb;

using CarShop.BusinessFacade;

namespace CarShop.DataAccess
{
    public class CarsGateway : GatewayBase
    {

        public CarsGateway() { }

        public void InsertCar(string make, string model, int
amount)
        {
```

```

Stringbuilder cmdText = new StringBuilder();
cmdText.Append("Select * from Cars where Make =
");
cmdText.Append(make);
cmdText.Append("");
cmdText.Append("and Model = ");
cmdText.Append(model);
cmdText.Append("");

OleDbDataReader reader =
base.GetReader(base.GetCommand(cmdText.ToString()));
if (reader.HasRows)
{
    reader.Close();
    cmdText.Remove(0, cmdText.Length);
    cmdText.Append("Select Stock from Cars where
Make = ");
    cmdText.Append(make);
    cmdText.Append(" and Model = ");
    cmdText.Append(model);
    cmdText.Append("");

    int stock =
(int)base.ExecuteScalar(base.GetCommand(cmdText.ToStrin
g())) + amount;

    cmdText.Remove(0, cmdText.Length);
    cmdText.Append("Update Cars set Stock = ");
    cmdText.Append(stock);
    cmdText.Append(" where Make = ");
    cmdText.Append(make);
    cmdText.Append(" and Model = ");
    cmdText.Append(model);
    cmdText.Append("");

```

```
base.ExecuteNonQuery(base.GetCommand(cmdText.ToString()));
    }
    else
    {
        reader.Close();
        cmdText.Remove(0, cmdText.Length);
        cmdText.Append("Insert into Cars (Make, Model,
Stock) values ('");
        cmdText.Append(make);
        cmdText.Append("'", "'");
        cmdText.Append(model);
        cmdText.Append("'", "'");
        cmdText.Append(amount);
        cmdText.Append(")");
    }
}
```

```
base.ExecuteNonQuery(base.GetCommand(cmdText.ToString()));
    }
}
```

```
public void LogSale(string user, string make, string
model, int amount)
{
    StringBuilder cmdText = new StringBuilder();
    cmdText.Append("Insert into Sales ([User], Make,
Model, Amount, [Date]) values ('");
    cmdText.Append(user);
    cmdText.Append("'", "'");
    cmdText.Append(make);
    cmdText.Append("'", "'");
    cmdText.Append(model);
    cmdText.Append("'", "'");
}
```

```
cmdText.Append(amount);
cmdText.Append(", ");

cmdText.Append(DateTime.Now.ToShortDateString());
cmdText.Append(")");

OleDbCommand cmd1 =
base.GetCommand(cmdText.ToString());

cmdText.Remove(0, cmdText.Length);
cmdText.Append("Select Stock from Cars where
Make = ");
cmdText.Append(make);
cmdText.Append(" and Model = ");
cmdText.Append(model);
cmdText.Append(")");

int stock =
(int)base.ExecuteScalar(base.GetCommand(cmdText.ToStrin
g())) - amount;

cmdText.Remove(0, cmdText.Length);
cmdText.Append("Update Cars set Stock = ");
cmdText.Append(stock);
cmdText.Append(" where Make = ");
cmdText.Append(make);
cmdText.Append(" and Model = ");
cmdText.Append(model);
cmdText.Append(")");

OleDbCommand cmd2 =
base.GetCommand(cmdText.ToString());

OleDbCommand[] cmds = new OleDbCommand[] {
cmd1, cmd2 };
```

```

        base.ExecuteNonQueryWithTransaction(cmds);
    }

    public void
    GetAllCars(System.Windows.Forms.BindingSource
    container)
    {
        try
        {
            OleDbDataReader reader =
            base.GetReader(base.GetCommand("Select * from Cars"));
            container.Clear();
            while (reader.Read())
            {
                Car car = new Car();
                car.Manufacturer =
            reader["Make"].ToString();
                car.Model = reader["Model"].ToString();
                car.Stock = (int)reader["Stock"];
                container.Add(car);
            }
            reader.Close();
        }
        finally
        {
            base.CloseConnection();
        }
    }

    public ArrayList GetAllCars()
    {
        ArrayList container = new ArrayList();
        try

```

```

    {
        OleDbDataReader reader =
base.GetReader(base.GetCommand("Select * from Cars"));
        container.Clear();
        while (reader.Read())
        {
            Car car = new Car();
            car.Manufacturer =
reader["Make"].ToString();
            car.Model = reader["Model"].ToString();
            car.Stock = (int)reader["Stock"];
            container.Add(car);
        }
        reader.Close();
    }
    finally
    {
        base.CloseConnection();
    }
    return container;
}

```

```

public ArrayList GetModels(string manufacturer)
{
    StringBuilder cmdText = new StringBuilder();
    cmdText.Append("Select Model from Cars where
Make = ");
    cmdText.Append(manufacturer);
    cmdText.Append("");
    ArrayList models = new ArrayList();
    try
    {
        OleDbDataReader reader =
base.GetReader(base.GetCommand(cmdText.ToString()));

```

```

        while (reader.Read())
        {
            Car car = new Car();
            car.Model = reader["Model"].ToString();
            models.Add(car);
        }
        reader.Close();
    }
    finally
    {
        base.CloseConnection();
    }
    return models;
}

```

```

    public void
    GetAllSales(System.Windows.Forms.BindingSource
    container)
    {
        try
        {
            OleDbDataReader reader =
            base.GetReader(base.GetCommand("Select * from Sales"));
            container.Clear();
            while (reader.Read())
            {
                SaleLog sLog = new SaleLog();
                sLog.Id = (int)reader["Id"];
                sLog.Manufacturer =
            reader["Make"].ToString();
                sLog.Model = reader["Model"].ToString();
                sLog.Amount = (int)reader["Amount"];
                sLog.User = reader["User"].ToString();
                sLog.Date = (DateTime)reader["Date"];
                container.Add(sLog);
            }
        }
    }
}

```

```

    }
    reader.Close();
}
finally
{
    base.CloseConnection();
}
}

```

```

public void
GetSalesWithinDate(System.Windows.Forms.BindingSource
container, DateTime from, DateTime to)

```

```

{
    StringBuilder cmdText = new StringBuilder();
    cmdText.Append("Select * from Sales where
([Date] >= #");
    cmdText.Append(from.ToShortDateString());
    cmdText.Append("#) and ([Date] <= #");
    cmdText.Append(to.ToShortDateString());
    cmdText.Append("#)");
    // cmdText.Append("");

    try
    {
        OleDbDataReader reader =
base.GetReader(base.GetCommand(cmdText.ToString()));
        container.Clear();
        while (reader.Read())
        {
            SaleLog sLog = new SaleLog();
            sLog.Id = (int)reader["Id"];
            sLog.Manufacturer =
reader["Make"].ToString();
            sLog.Model = reader["Model"].ToString();
            sLog.Amount = (int)reader["Amount"];

```

```

        sLog.User = reader["User"].ToString();
        sLog.Date = (DateTime)reader["Date"];
        container.Add(sLog);
    }
    reader.Close();
}
finally
{
    base.CloseConnection();
}
}

```

```

public ArrayList GetSalesWithinDate(DateTime from,
DateTime to)

```

```

{
    ArrayList container = new ArrayList();
    StringBuilder cmdText = new StringBuilder();
    cmdText.Append("Select * from Sales where
([Date] >= #");
    cmdText.Append(from.ToShortDateString());
    cmdText.Append("#) and ([Date] <= #");
    cmdText.Append(to.ToShortDateString());
    cmdText.Append("#)");
    // cmdText.Append("");

    try
    {
        OleDbDataReader reader =
base.GetReader(base.GetCommand(cmdText.ToString()));
        container.Clear();
        while (reader.Read())
        {
            SaleLog sLog = new SaleLog();
            sLog.Id = (int)reader["Id"];

```

```

        sLog.Manufacturer =
reader["Make"].ToString();
        sLog.Model = reader["Model"].ToString();
        sLog.Amount = (int)reader["Amount"];
        sLog.User = reader["User"].ToString();
        sLog.Date = (DateTime)reader["Date"];
        container.Add(sLog);
    }
    reader.Close();
}
finally
{
    base.CloseConnection();
}
return container;
}

```

```

public ArrayList GetAllSales()
{
    ArrayList container = new ArrayList();
    try
    {
        OleDbDataReader reader =
base.GetReader(base.GetCommand("Select * from Sales"));
        container.Clear();
        while (reader.Read())
        {
            SaleLog sLog = new SaleLog();
            sLog.Id = (int)reader["Id"];
            sLog.Manufacturer =
reader["Make"].ToString();
            sLog.Model = reader["Model"].ToString();
            sLog.Amount = (int)reader["Amount"];
            sLog.User = reader["User"].ToString();
            sLog.Date = (DateTime)reader["Date"];

```

```
        container.Add(sLog);
    }
    reader.Close();
}
finally
{
    base.CloseConnection();
}
return container;
}
}
}
```

UsersGateway.cs

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Data.OleDb;

using CarShop.BusinessFacade;

namespace CarShop.DataAccess
{
    public class UsersGateway : GatewayBase
    {
        public UsersGateway()
        {
        }
    }
}
```

```
public void InsertUser(string userName, string
password)
{
    StringBuilder cmdText = new StringBuilder();
    cmdText.Append("Insert into [Users] ([UserName],
[Password]) values ('");
    cmdText.Append(userName);
    cmdText.Append("'", "'");
    cmdText.Append(password);
    cmdText.Append(")");

base.ExecuteNonQuery(base.GetCommand(cmdText.ToStrin
g()));
}
```

```
public void ValidateUser(User user)
{
    StringBuilder cmdText = new StringBuilder();
    cmdText.Append("Select * from users where
UserName = '");
    cmdText.Append(user.UserName);
    cmdText.Append("");
    cmdText.Append("and Password = '");
    cmdText.Append(user.Password);
    cmdText.Append("");

    try
    {
        OleDbDataReader reader =
base.GetReader(base.GetCommand(cmdText.ToString()));
        if (reader.HasRows)
            user.Validated = true;
        reader.Close();
    }
}
```

```
finally
```

```
{
```

```
    base.CloseConnection();
```

```
}
```

```
}
```

```
}
```

```
}
```