

**DESIGN AND IMPLEMENTATION OF A POULTRY FARM
DATABASE MANAGEMENT SYSTEM.
(A CASE STUDY OF CHICKEN –RUN POULTRY FARM)**

BY

OKOLO, FRANCES AWELE OMERUGA

REG NO: 01253135

**BEING A PROJECT SUBMITED IN PARTIAL FULFILMENT OF
THE REQUIREMENT FOR THE AWARD OF BACHELOR OF
SCIENCE DEGREE (B.SC) IN COMPUTER SCIENCE,**

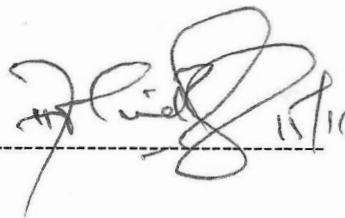
**DEPARTMENT OF MATHEMATICS, STATISTICS AND
COMPUTER SCIENCE, FACULTY OF SCIENCE, UNIVERSITY OF
ABUJA, ABUJA.**

NOVEMBER 2006.

CERTIFICATION

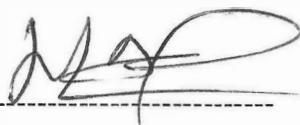
This is to certify that this project was carried out by Okolo, Frances Awele O. with Registration Number 01253135 in the Department of Mathematics, Statistics and Computer Science, Faculty of Science, University of Abuja, in partial fulfillment of the requirements for the award of Bachelor of Science degree (B.Sc) in Computer Science

Dr. M. B. Hammawa
Project Supervisor



Date 15/11/2006

Professor M. O. Adewale
Head of Department



Date 16/11/06

External Examiner -----

Date -----

DEDICATION

This project is dedicated to my parents Mr. and Mrs. F. A. Okolo, and my siblings Billy and Mandy, whose unconditional love and support I can always count on even in difficult times and when absolutely unnecessary has brought me thus far.

ACKNOWLEDGEMENT

I want to thank God for his infinite mercy, guidance and protection and for granting me the knowledge to accomplish this project and also to members of my family, nuclear and extended.

I also want to say a big ‘Thank you’ to the Head of Department, Professor Adewale, whose fatherly assistance we all could always count on, my project supervisor, Dr. Hamawa, for his time, patience and assistance in the completion of this project and finally all my lecturers for their tutoring, training and educational influence.

My gratitude to Adiya Atuluku, Thelma Thani, Moyo Lawal, Farouk Ahmed, Chimdi Anyama for their camaraderie, and Dubem Onyejiaka for his friendship.

Finally I want to thank Nedu and Hilary for their infallible assistance in the execution of this project.

ABSTRACT

This project was undertaken to investigate a safer and more reliable way poultry farmer may collect, record, manage and easily access data or information about the day-to-activities on a poultry farm. The software designed would assist the user in providing adequate financial statements of the poultry business. The software may be used by small to medium scale farmers in general.

TABLE OF CONTENT

Title page	i
Certification	ii
Dedication	iii
Acknowledgement	iv
Abstract	v
Table of Content	vi

CHAPTER 1 : Introduction

1.0 <i>Introduction</i>	1
1.1 Background of Study	4
1.2 Aim and Objective of project	4
1.3 Research methodology	5
1.4 Scope and limitations of the project	5

CHAPTER 2: Literature Review

2.0 Poultry Farming in Nigeria	6
2.1 Data Management in Poultry Farms	7
2.1.0 Poultry Farm Records	8
2.1.1 Reasons for Record keeping	9
2.1.2 Types of Poultry Records and Reports	11

CHAPTER 3: System Analyses and Design

3.0 <i>Investigation of current data management system</i>	12
3.1 Review of existing data management system.....	12
3.2 Proposed data management system over existing system.....	13
3.3 System design	14
3.4 Design Overview.....	15

CHAPTER 4: System Implementation

4.0	Structure of the application	18
4.1	Working Requirements.....	20
4.2	Soft wares used in program	20
4.2.0	Programming language	21
4.2.1	Database Management system	21

CHAPTER 5: Summary and Conclusion

5.0	Summary	22
5.1	Limitation	22
5.2	Recommendation	23
References	24
Appendix A	
Appendix B	
Appendix C	

CHAPTER 1

1.0 INTRODUCTION

Poultry Farming is the commercial raising of chickens, fowl, turkeys, geese, and ducks for their meat and eggs. The birds are kept intensively in a confined area where they are fed and managed carefully. Their economic significance varies considerably around the world.

Poultry farming in Nigeria is a very economically and commercially viable business due to the fact that poultry birds do not just serve as meat, but also produce eggs for human consumption and other purposes. Almost 8 of every 10 Nigerians are involved in agriculture and about 10% of the Nigerian populations are engaged in poultry production, mostly on subsistence, small, or medium-sized farms.

In Nigeria, the production of food has not increased at the rate that can meet the need of our increasing population.

Besides Nigeria's agricultural sector not being able to meet the food requirements of the country's constantly increasing population, its greatest problem is the inadequate animal protein in the diet of a large portion of the population especially the rural areas which make up 60% of the Nigerian population. It has been reported that the average Nigerian takes in less than 20% the average amount of animal protein required. This animal protein is found in most poultry products. (Titilola 1990)

Eggs are very profitable for the poultry farmers. They are very nutritious and serve as a high source of animal protein.

Eggs are more easily affordable by the common man than other sources of animal protein. An average boiled egg costs N20, hence boiled eggs are easily attainable.

Chicken Eggs serve as a major component in many recipes for meals and confectionaries such as bread, cakes, ice cream and so much more. This brings about high and continuous demand for the eggs.

The droppings of these poultry birds produced on poultry farms are collected and sold in bags to crop farmers who use them as manure. The sale of these droppings also serves as a source of revenue to the farm. Thus, the business of poultry farming can not

be underestimated due to the numerous transactions carried out on the farm. It is therefore important to have an adequate data management system.

In every business, small or large scale, accurate information is paramount to its management and success because it is important in decision making. Therefore, it is important that the method, through which this information is collected, stored, organized and managed, is reliable and secure. Accurate information can be said to be data that is correct and is declared true based on certain criteria. A data management system is a system designed to handle and manage data collected in a business or organization. It provides a way to store data for future access and use.

In small scale Poultry Farming, data management systems are very poor and inadequate. Farm records pertain to information recorded on the day-to-day operation of a particular farm. Records are statements of fact or data concerning a specific subject which may be specified in physical, monetary, mathematical, or statistical terms.

A farmer should keep farm records because it is important for him to know what his farm is worth at any point in time. He should also be able to know how much the business has made over a period of time. Proper record keeping helps the farmer obtain at a very short notice the types of information he may need for managerial decisions. (Duvick, 2004). It helps in planning and budgeting the farm operation. Farm records kept over a long period of time can help the farmer in describing the history of the farms' performance and therefore provide a basis for making comparisons with other similar farms.

Good record keeping helps the farm manager to control current operations and check specific arms of the business from time to time to see if the farm is being operated in line with the plans and set goals.

The most commonly used system for data management is filing cabinets filled with paper records. In many cases, the data stored for these poultry farms are in record books. Data stored in record books are usually done by hand or using typewriters. Hand written data have many problems due to varying hand writings which are often illegible. Books get torn very easily. They can easily be soiled or even destroyed. These could easily bring about loss of data. Critical data stored on writing materials can easily be

stolen or altered by unauthorized persons if it falls into the wrong hands. Often, other records in Poultry farms are not even documented.

There is also, the problem of data retrieval. Data retrieval is the process of accessing information that has been stored. Data kept in record books over a long period of time are often poorly organized and sometimes unquantifiable. Finding records or accessing data is extremely difficult and almost impossible. Sorting this quantity of data is absolutely unachievable and even when sorting is finally done; the results are very unreliable due to numerous human errors. Also these paper filing systems generally does little or nothing to show the relationships between records.

The lack of a proper data management scheme cannot be over emphasized. It could lead to innumerable consequences which can grossly affect the progress of the poultry farming. In fact inaccurate or inadequate information in poultry farming can lead to a colossal loss or total collapse of a business.

It is obvious then, that in the Poultry Farming business, a data management system is very important to keep track of the various activities that are carried out on the farm. Most of these activities include:

- Purchase of birds, feeds e.t.c, which amount to expenditure must be taken in to account to get a comprehensive statement of account on the Farms financial status.
- Taking records of daily production quantities to ascertain and monitor the productivity level of the poultry Farm.
- Keeping records of daily sales to understand the market demand and supply.
- Bills such as water, light, medical, transportation costs and other expenditure must be taken in to consideration.
- *Information concerning the employees and their background* is collected on employment of the employee for reference in future.
- Record of fraud or theft on the poultry
- Staff duty roster.

In all, extensive information and data gathering help to give the farmer a clear, explicit and accurate perspective on the status of the farm pertaining to the productivity level of the birds and the financial aspect of the poultry.

A computerized database management system is the solution to all the problems listed. These are software systems, usually automated for the management of any collection of compatible data, which provide a way of collecting, organizing, manipulating and storing data. They provide security for these vital information by limiting access to only authorized users and ensuring data accuracy and accessibility.

1.1 BACKGROUND OF STUDY

The case study I intend to use for this project is Chicken-Run poultry Farms situated in Delta state. The poultry is a medium scale poultry farm housing up to three thousand birds. These include layers, broilers, cockerels and turkeys.

The poultry was established in 2001 and started off with about five hundred birds. Today, the poultry is more than twice its original size. The poultry raises birds for sale. Most of the activities carried out by the farm include mainly Production and sales of eggs and droppings.

Since its establishment in 2001, the farms' production, Sales, expenditure e.t.c have increased considerably. The use of papers and file to store data is no longer adequate for the poultry farm record keeping. I intend to design a computerized data management application that will meet the growing data management needs of the Poultry farm.

1.2 AIMS AND OBJECTIVES

The aim of this project is to design computerized database software which is very easy to use, organizes, manages and maintains the data collected on the Poultry farm. It will provide the user with data security, back up facilities and data report with respect to data stored.

1.3 RESEARCH METHODOLOGY

Research methodology is the way in which the data are collected for the research project. Research can be done in different ways which include using Questionnaires, interviews, libraries with records of already collected data or by observation or participatory methods.

The research methodology I intend to apply involves interviewing the poultry manager and other workers in the poultry farm. Research has shown that the highest level of accuracy in information collected is obtained from interviews and participatory methods of research. The question and answer sessions will assist me in designing the application to meet the requirements of the poultry.

I also intend to use the observatory method and take part in the activities on the farm so I can acquire first-hand knowledge about the activities that go on in the farm and apply this knowledge to the creation of the software.

1.4 SCOPE AND LIMITATIONS

The scope of this project is based on the case study (Chicken-Run Poultry Farm, Delta State). The limitations of the project will depend on the parameters of the case study.

Chicken – Run Poultry Farm is a medium scale poultry farm that houses over three thousand birds for sale, which are limited to mostly Layers, broilers, cockerels and turkey. It handles egg production as well as droppings production. The business does not include production of hatchling. Young birds are purchased and grown to market size for sale. Feed for the poultry birds on the farm is purchased and not made on the farm.

The software will be designed to meet the needs of small - medium scale poultry farming as will suited the case study.

CHAPTER 2

LITERATURE REVIEW

2.0 POULTRY FARMING IN NIGERIA

About 80% of the Nigerian population are involved in Poultry farming. Most of these are in subsistence, small scale and medium scale farmers. The two main Poultry management systems in Nigeria are extensive and intensive.

The extensive or scavenging system has two subsystems: The backyard system, also called the family or subsistence system, in which a small number of birds are partly confined within a fenced yard, and the free-range system, also called the traditional or Yarding system, in which birds are free to roam around the homestead. The intensive system also includes two subsystems: the semi-intensive system, in which a number of birds are produced in complete confinement, and the battery cage system.

Nigeria's poultry production is expanding but is not keeping up with rapidly increasing domestic market. Nevertheless, the industry possesses enormous capacity for growth in the medium to long-term.

Research has showed that there has been an increase in the demand for poultry products over the past 3 years (after more than a decade of decline). This is mainly due to an increase in the fast food restaurants which feature chicken menu in the major urban area and growth in the sales through institutional catering facilities. (Ali, 2002)

Chickens make up the majority of all poultry consumed in Nigeria. Per Capital chicken consumption for Nigeria's populations does not exceed 1 kilogram, which is equivalent to about two-third of a chicken. (Ojo, 2003). A characteristic of the poultry meat market is the varying preference among income earners. The poultry meat is very tender so it is palatable and acceptable to many consumers causing high demand. Low-income Nigerians prefer the tough spent hens for laying operations, while the higher income and the expatriate communities prefer the tender broiler meat. This preference in taste results in variation in the market prices.

Poultry meat is regarded as a delicacy and very often eaten during celebrations and festivities. This brings about increase in price during festive seasons.

The per capital consumption of Chicken Eggs is estimated at 20, reflecting the negative attitude in traditional societies which regard egg consumption as a luxury. Chicken Egg protein has a biological value of 1.0 and so shares with human protein the distinction of being a perfect protein.

The droppings of these poultry birds are collected in bags. It is dried and sold to crop farmers who use them on their farms as manure. The sale of these droppings serves as a source of income for the farm. The quality and quantity of the excrement depends on the feed that is given to the birds. The droppings or excrement of poultry birds are rich in nitrogen and phosphate which is good for farm crops like maize and beans. Chicken excrement is particularly high in Nitrogen and phosphates compared to other birds.

Poultry farming has the following advantages over other live stocks;

- Poultry birds are good converters of feed into good protein in meat and eggs. Animal protein is essential in human nutrition because of its biological significance.
- The production cost per unit is low relative to other types of livestock and the return to invest is high, so the farmer needs only a small amount of capital to start a poultry farm.
- It has a short production cycle through which the capital is not tied down over a long period of time.

2.0 DATA MANAGEMENT IN POULTRY FARMING

In all spheres of life, adequate information is necessary for decision making. This is also the case for farming decisions. The farmer needs information on a number of things before he can take appropriate decisions. In taking appropriate decision on whether to purchase an input, say feed, he needs to be informed about the types available, their cost prices, which centre nearest to him they can be obtained, and the cost and means of transportation. Likewise, when he wants to dispose of his produce, he needs adequate market information.

Before a decision is taken, it is very important to gather the necessary information. The farmer can gather this information from neighbouring farms, extension agents, research institutes, government organization, commercial concerns and his own

records. A very important source of information is the farmers' own documented records. However due to illiteracy, the traditional farmer does not keep records of his past operations in a written form, though there is enough evidence to suggest that he does use the experience he has acquired over the past years.

2.1.0 Poultry Farm Records

Records are statements of fact or data concerning a specific subject which may be specified in physical, monetary, mathematical and statistical terms. Poultry Farm records pertain to information recorded on the day-to- day operation of a Poultry farm. For a specified period of time, the records may show the various types of assets owned, their purchase prices, dates of purchase and the conditions at purchase. Information on inputs (durable and non-durable) would not only be recorded but the output situation too will be recorded.

A complete poultry farm record would include all the daily activities and transactions and with a proper accounting system. It should be able to have a complete estimate of the profit or loss statement at the end of the year.

By keeping a complete farm record on a yearly basis, it is possible to determine the growth or deterioration of the farm.

The information required for adequate record keeping are;

- (1) Labour: The daily records of labour should be recorded. The name, age, sex, type of work done, and number of hours spent by each category of worker needs to be kept. The cost of hired labour (where necessary) should be kept.
- (2) Supplies: The poultry farm supplies such as feed, litter, disinfectants, drugs and vaccines would be recorded. This is done by type source, cost and quantities used by date.
- (3) Output: The total number of eggs collected on a particular day, condition of the eggs (cracked or not fully formed) and the sizes must be carefully recorded.
- (4) Sales and Marketing Cost of Poultry Products: for each type of product sold, the record should include the name, type and number of units of the produce sold, place of sale and the revenue received. The mode and cost of transportation to the place of sale will also be recorded.

(5) Price: the farm gate prices should be kept in addition to the retail prices in the local market.

2.1.1 Reasons for Record keeping

A poultry farmer should keep records because it is extremely essential for him to know what the farm is worth at any point in time. He should also be able to know how much a business has earned over a period of time. Proper record keeping helps the farmer to obtain in a short time the type of information he may need for managerial decisions. It helps in planning and budgeting of the poultry operations. (Sturrock, 1971)

Good record keeping helps the farmer to control and check specific aspects of the poultry farm at intervals.

Farm records may be required by certain government agencies for the establishment of development schemes and setting up of product controls. A concern of many financial consultants and lenders is that many farmers lack adequate records to monitor and analyze the financial health of their farm business. Lack of financial data makes it difficult to determine proper debt loads, or to develop plans for businesses with major financial problems. With the establishment of agricultural credit guaranteed schemes, it is necessary for farmers to produce records of their farm activities to determine their suitability for credit (Castle et al, 1972).

Another reason most farmers keep records is for tax reporting. It is possible to determine the amount of tax an individual poultry farmer should pay by analyzing the Poultry farm records. However, it may be self defeating to disclose this reason to the farmer when convincing them to keep records because traditional farmers would normally shy away from any thing that will make the government assess them on a higher tax level. Nevertheless, accurate records remain the authentic source for estimating a poultry farms' appropriate tax. Tax records get even more complex when the business has employees.

Availability of accurate farm records aids the research process. Research in agricultural economics in less developed countries always appears more cumbersome and time, energy and fund consuming than in developed economies. This is because majority of the farmers are illiterate and can hardly keep their own poultry records. Farm

management surveys therefore need a system of instituted questionnaires which can be administered through frequent interviews. This method has allotted some problems concerned with memory lapse on the part of the farmer as well as delays in completing the analysis for any useful intervention on the poultry farm. Such problems can be minimized if farm records are readily available or can be obtained through e-mail and telephone calls as in developed countries (Castle et al, 1972).

In addition to enhancing research, poultry farm records aid teaching, and form the basis for developing teaching materials for students of Agriculture. This is why at the university level, teaching and research go hand-in-hand to ensure that lecturers remain current in their fields.

Lastly, farm records forms the basis for government policies. Pricing policies which are sometimes geared towards aiding the farmers cannot be done when there is paucity of data. Market schemes can succeed only if based on accurate data from farm records.

As poultry farming moves from subsistence to commercialization, records become more and more important to farmers, teachers, researchers, extension agents and all. The details and amount of data to be kept will vary according to its intended purpose. In summary, poultry farm records are necessary because:

- They give a measure of the poultry's success.
- Most economic research depends on some knowledge of accounting procedures such as budgeting, cost studies, and modern programming techniques which rely on farm records.
- They give guidance in making farm production recommendations which are based on cost and return of production.
- Resettlement scheme policies should be based on costs and returns to determine appropriate organizations.
- Marketing, pricing, and development policies should be related to costs and returns of poultry production derivable from farm records.
- Successful teaching at all levels of agricultural education depends on good farm record keeping.

2.1.2 Types of Records

The various types of records to be kept include Farm inventory records, income and recipes records, the livestock expense records, farm labour records, and net farm profit records.

An accounting system is simply term for farm records. Users should think about four types of data needed for the Poultry farm business accounting system:

1. Daily cash farm receipts and expenses, including loans and payments.
2. Capital purchases and sales.
3. Annual inventory at end of the business year.

2.1.3 Types of Reports

To report how the Poultry farm business is doing requires at least two financial statements: the Balance Sheet and the Income Statement.

The balance sheet shows the farmer what he owns and owes. The balance sheet is helpful in acquiring debt, measuring financial progress, considering risk and valuing ownership. The income statement shows what the business earned during the business year. This should be an accrual statement. A true income statement will reflect what the poultry actually produced during the year. Other reports include Sales reports, Expenditure report, and Cash Flow in and out of the poultry business.

CHAPTER 3 **SYSTEM ANALYSIS AND DESIGN**

3.0 Introduction

Chicken Run Poultry farm uses the file cabinet system in managing data on the farm. Data is recorded using record books. There is the Sales book. The Sales book is used to record all the sales on the farm. These include eggs, droppings and mostly poultry birds. The book is ruled in to rows and columns. The columns include the date of the sale, the item, the quantity purchased, the unit price and the total amount paid. There is no record of the customer who purchased the goods or the customers address for future reference. At the end of the day, the total sales are calculated and recorded.

There is also an expenditure book where expenditure is recorded. These expenditures include hatchlings, feed, vaccines, labour, electricity and water bills e.t.c. Here also the wages and salaries of the worker are recorded when they are paid. Important activities like Debeaking (trimming the beaks of the birds to prevent cannibalism), vaccination against common diseases, deaths of birds and other things of great significance are usually not recorded or at best written on a calendar. There is no record of the production of eggs or droppings. This does not do much to help the farmer monitor the performance of the birds.

3.1 REVIEW OF EXSISTING DATABASE MANAGEMENT SYSTEM

The current data management system employed by Chicken-run poultry Farm is grossly inadequate for a good number of reasons.

To begin with, this system does little or nothing for data security. If any of the books are misplaced or stolen, all the records are totally lost. For instance, if the current sales book is misplaced or lost for any reason, all the sales recorded in that book are lost. It does not do much for back up since no farmer keeps two of the same record books.

Often, the records stored in record books are usually done by hand. Information written by hand, as stated before, has many problems due to varying hand writings which may be illegible. As the record keepers change, hand writings change making understanding the record stressful due to lack of consistency. Depending on the literacy

level of the record keeper, spelling mistakes, grammatical errors, omissions and other errors abound.

Record books are very easily mutilated due to daily use. They wear and very often, fall apart with age. They can easily be destroyed bringing about loss of data. Records kept on calendars are visible to all including unauthorized personnel. This is very detrimental to the business especially in cases of fraud. The process of retrieving data from these record books is very difficult and with time, as data increases, this will become impossible. The data organization in these record books makes it difficult to analyze and evaluate the records and state of affairs. Where evaluation is done, it is based on many assumption and errors. Since no particular record on the employees are kept, where there is a case of theft or fraud, locating the employee, if one is responsible is not very easy. The data management system I am proposing solves most of, if not all the problems stated above.

3.2 PROPOSED DATABASE MANAGEMENT SYSTEM

The system I am proposing is database application which is called MicroFarm. It is computerized. All the farm data are stored directly in the computer. This ensures that the farm data cannot be easily lost, mutilated or destroyed.

Micro farm provided data security by providing the user with the password ability. This prevents unauthorized users from accessing vital data on the system. Even in a case where the password is leaked to the wrong persons, MicroFarm is designed to provide the system user with the ability to change the password and re-establish sole access and control.

MicroFarm is design with a relational database system which ensures that data is stored in a tabular format to promoted organization and easy access of data from the system. Better still, MicroFarm is very easy to use. Since most farmers including poultry farmers are novice when it comes to the use of the computer, MicroFarm has been designed to provide a very friendly graphic user interface to promote and encourage its use. Technical argons and “big” words which may confuse the not-so-literate farmer have been avoided and at best kept to the barest minimum.

MicroFarm attempts to minimize human errors enter into the system such as numbers instead of alphabets by implementing constraints and checks on data entered and providing error messages to alert the user about the error.

MicroFarm allows the user to record new profiles, either of employees or customers in order to keep track and records of them. It allows the user to record and save vital information like the name, address, phone numbers and allot more about the employee or customers. MicroFarm also allows the user to save pictures of these people.

The software also provides the user with quick and easy access to these profiles (employees and customers). By entering the employee code, customer code, first name or last name, MicroFarm accesses the record and displays it on the screen.

MicroFarm is designed provides the farmer with quick access to reports. These reports based on the date specified by the user. This may be daily, weekly, monthly, or even yearly. These reports are printable and accurate depending on data entered into the system by the user.

3.3 SYSTEM DESIGN

The design process will attempt to incorporate desirable programming practices which include data security, back up and recovery, efficiency and most especially ease of use (user friendliness).

The design process will involve two aspects; one of which will deal with the deal with the database aspect of implementation and the other the Interface of the software. The database aspect, which is the back end of the system, will basically deal with the data management needs of the software. The database will be made up of the various tables which will organize, store and manipulate the data enter by the user. The User Interfaces aspects will basically be responsible for the interaction between the system and the user. It will deal with inputs made by the user and will provide constraints on these data to ensure that the information entered into the system is as accurate as possible.

The finished software ought to be such that there will be a security feature, the login process which will entail a password to ensure the security of the software and the data stored within it.

To ensure that the graphic user interface is user friendly, simply English terms will be used. Non technical jargons will be avoided and tool tip text will be incorporated so as to enable the users identify the various sections of the software and activities they carry out. The software will be design to look like regular Windows applications and will have the menu bar to ensure ease of use of the software.

The user input to the software (MicroFarm) will be done using the mouse and the key board. The design will ensure that the user can navigate the application using the keyboard although the mouse can also be used in the input process.

All the reports produced by the system will be displayed such that they can be printed if necessary. Since the one main essence of the software is to provides quick and easy access to data stored, the output will be such that all input stored can be reviewed in the reports. These outputs include the Sales Report and the Expenditure Report.

3.4 DESIGN OVERVIEW

The system is designed to be as simple as possible so that poultry farmers who are not well learn in the use of the computer system are not confused and discourage by its complexity. Therefore, there are no elaborate features or there are very few. However, the features used here basically include (as stated in system design) the menu bar and easily understandable icons which provide a user friendly graphical interface.

The user interface has attractive and familiar pictures to keep the user comfortable and tool tip help to assist the user.

Messages are displayed on screen and an alarm sounded when data is about to be saved or when an error has been made by the user possibly while entering data into the system.

The user input to the software will be done using the mouse and the keyboard. The design will ensure that the user can navigate the application using the keyboard although the mouse can also be used in the input process

The system is specially designed to detect errors especially when data is incorrectly entered into the system like numbers instead of alphabets. In the case of entering the data and other error prone situations, special date tools have been put in place to ensure that those errors are avoided as much as possible.

The structure of the database is very simple since it will entail only the storing of data on the poultry farm which is not very large when compared to data stored by larger organisations such as banks and telecommunications companies.

As shown in the design overview in chapter three, the database is made up of eight entities (tables) which include:

- Employee: this keeps the records of the various employees who work on the farm.
- Customers: this keeps the records of the customers that purchase the farm products
- Sales: this records all the sales of the farm products which may be birds, eggs or droppings.
- Expenditure: this keeps the record of all the purchases, salaries and wages and miscellaneous expenditure on the farm.
- Password: this stores the system access username and password
- Activities: this keeps track of all the activities on the farm which may include vaccination, Debeaking, deaths of birds, illnesses e.t.c.
- Egg Production: this keeps record of the number of Eggs produced per farm house on a daily basis.
- Droppings Production: This stores record of the amount of droppings produced on a daily basis.

The relationships between the above stated entities have been explicitly stated in the entity relationship diagram (ERD). (Fig 3.1) And the processes that make up the software functionalities and data processes are explicitly described in the data flow diagram. (See Appendix Fig: 3.10)

ENTITY-RELATIONSHIP DIAGRAM (ERD)

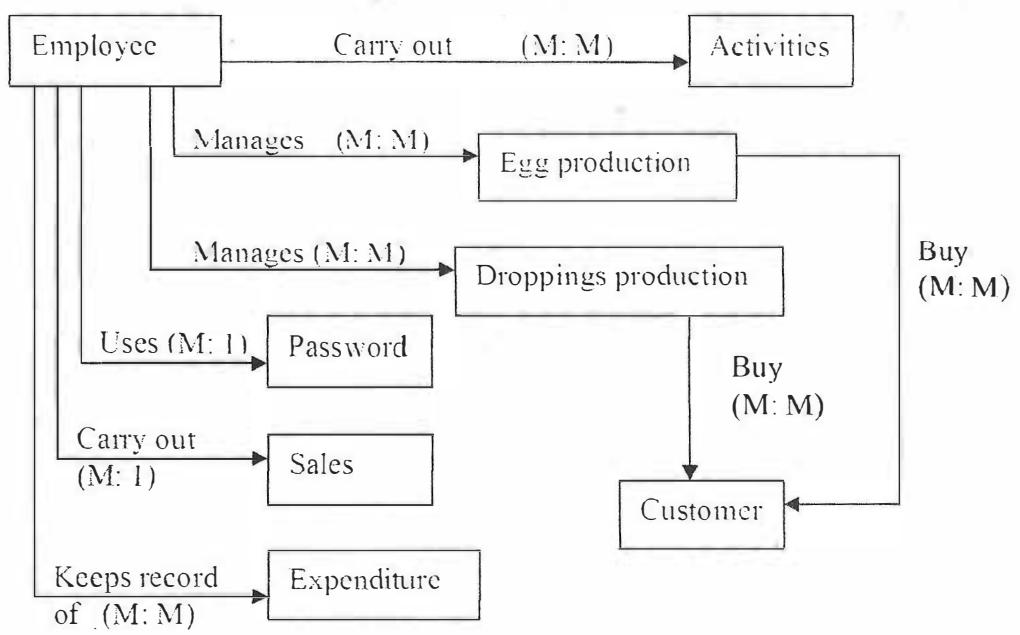


Fig 3.1

CHAPTER 4

SYSTEM IMPLEMENTATION

4.0 Structure of the Application

When the icon representing the software is clicked the splash screen introducing the software is displayed. After 5 seconds or double-clicking on it, the login section is displayed.

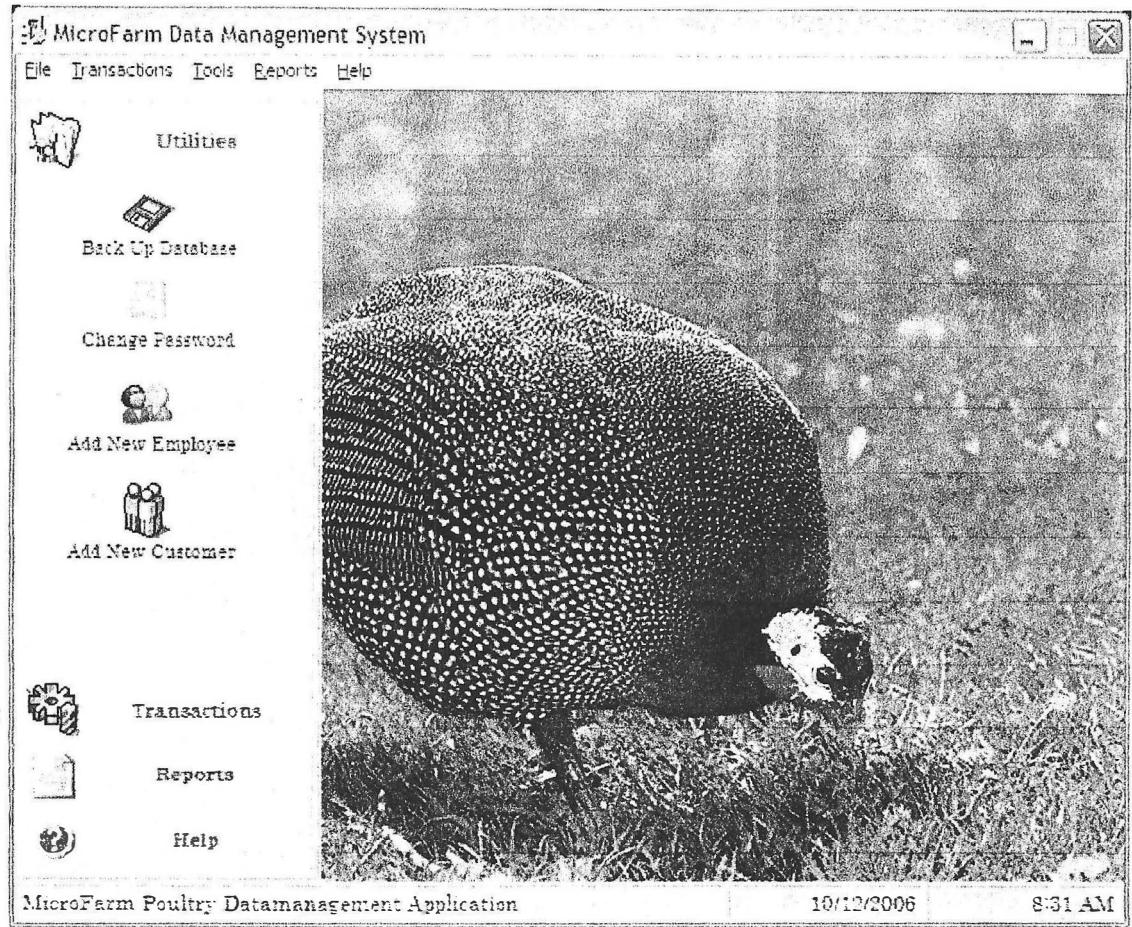
Fig 3.2: Splash Screen



The login section is designed to accept a username and password which will grant the user access to the main part of the software otherwise, on entering the wrong password it will deliver an error message. On entering the wrong password twice, the system will terminate. The system may also be terminated at this stage by pressing the “Esc” button.

The main window, which can be minimized, consists of the menu bar and icons for the various software functionalities. The menu bar has File, Transactions, Tools, Report and Help. On clicking any of these menu bar items, a drop-down list containing various functionalities is displayed. These functionalities possess short-cut icons on the main window.

Fig 3.3: Main window

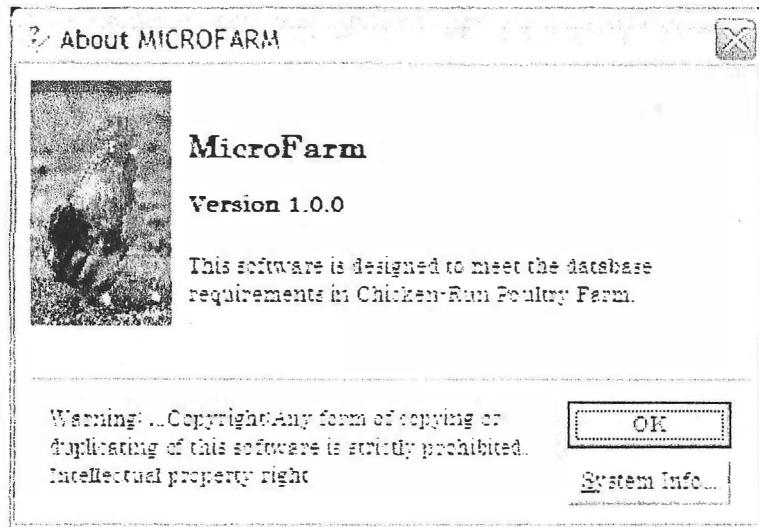


The File list displays “Add New Records”, “Add New Profile”, “Find Profile” and “Exit”. The “Add New Record” allows the user to add new records to the database. These records may be “Activity”, “Egg Production” or “Droppings Production”. The “New Profile” allows the user to save new details on either employees or customers. The “Exit” allows the user to terminate the system. It asks the user to ensure he/she wants to exit and gives the user the opportunity to go back instead of exiting the system.

The “Transactions” list displays “Sales” and “Expenditure”. The Sales module allows the user to enter and store the daily sales and the Expenditure module allows the user to enter and save all the expenditure. The “Tools” drop-down menu allows the user to change the system password and the database option of backing up the system. The “Report” drop-down menu allows the user to create Egg or dropping Production reports, or Sales and Purchases report depending on the date chosen by the user. The “Help” drop

down menu provides the user with information about the system. The version number and the copy right laws associated with it. It also allows access to the system Information.

Fig: 4.



See Appendix B for Output Results

4.1 SYSTEM REQUIREMENTS

The working requirements of the software include:

- 15" VGA monitor
- Printer
- Keyboard and mouse
- Windows operating system(Windows 98, 2000, XP Professional)
- Celeron or Pentium 2 and higher 3000Mhz
- SDRAM of 256MB

4.2 SOFTWARES USED IN PROGRAM

The softwares for the system design include Microsoft Visual Basic 6.0 and Microsoft Access. These are used together to create a client server application where Microsoft Visual Basic 6.0 is used to design the front-end and Microsoft Access is used for the back-end.

4.2.1 PROGRAMMING LANGUAGE

Microsoft Visual Basic 6.0

Microsoft Visual Basic 6.0 (also commonly known as VB) lets you write, edit and test windows applications. In addition, Visual Basic 6.0 includes tools to write and compiles help files, ActiveX controls and even Internet applications.

Microsoft Visual Basic 6.0 is a language rapid application development environment that gives you fast, easy, and intuitive tools to quickly develop Windows applications. Using Visual Basic, you can develop simple utilities or sophisticated applications. Data access features allow you to create databases, front-end applications, and scalable server-side components for most popular database formats. Visual Basic programming language is fairly simple to use and understand. It uses common English words and phrases for the most part.

4.2.2 DATABASE MANAGEMENT SYSTEM

Microsoft Access is a relational database management system from Microsoft. Microsoft Access can use data stored in Access, Microsoft SQL Server, Oracle, or any ODBC-compliant data container. Skilled software developers and data architects use it to develop powerful, complex application software. Access is widely used by small businesses, within departments of large corporations for handling the creation and manipulation of data. Its ease of use and powerful design tools give the non-professional programmer a lot of power for little effort.

One of the benefits of Access from a programmer's perspective is its relative compatibility with SQL— queries may be viewed and edited as SQL statements. Users may mix and use both Visual Basic Application for programming forms and logic and offers object-oriented possibilities.

CHAPTER 5

5.0 SUMMARY

The database management software MicroFarm is designed to meet the data management needs of the poultry farmer. It takes into consideration all the various activities carried out on the poultry farm and provides the farmer with adequate storage and easily access to data and also reports.

The software is especially designed to ensure simplicity and ease of use. It provides the user with a simple, interactive graphical user interface. It begins with the login system which enforces data security using the username and password for access, and leads to the Main window, where the user or farmer can choose what he/she would like to do.

It allows the user to keep record of the daily activities on the poultry, the number of eggs and droppings produced, and also to save profile and details about employees who work on the poultry farm, and the customers who patronize the farm and also allows for the saving of daily transactions on the poultry, which are mainly expenditure and sales of products. It provides back-up for data in case of data loss and also the ability to change username and password.

Finally, it provides the user with Sales and Expenditure report to improve the financial analysis of the poultry business, and Egg and Droppings production report to give the farmer a clear picture of the farms productivity.

5.1 LIMITATIONS

To begin with, the constraints of this project are few. First the software is designed to meet the needs of the case study. These needs may vary a little from one poultry farm to another. It is limited to the used of small to medium scale poultry farms in Nigeria.

The software does not provide the user with a variety of reports and statements which include Balance sheet and Income statement. It is currently limited to four different reports. Also, it does not provide its users customers with proof of purchases (Receipt) for the products bought.

Finally, the system is limited to the use of only one username and password which provides access to the entire application. There are no levels of access. It does not provide multiple user account for varying levels of data access. Once a user accesses the system, he/she has full access to all the data stored.

5.2 RECOMMENDATION

My recommendation for this project basically include what effort could be made to overcome the stated limitations so as to produce an extremely durable, effective and generally acceptable software.

The system could improved to be able to provide a variety of reports and statement which should include Balance sheets and income statements. Also, it should be able to provide receipts for farm sales as proof of purchase. These are to improve is usability and efficiency.

Also, a chart report such as line graphs, bar graphs or pie chart e.t.c. could be implemented to provide the user with a mean of evaluating the production, sales e.t.c at a glance. This will improve the ease of use of the software. Finally, a multi user access level such as Administrator guest user, e.t.c. could be designed to provide limited access to certain personnel.

REFERENCES

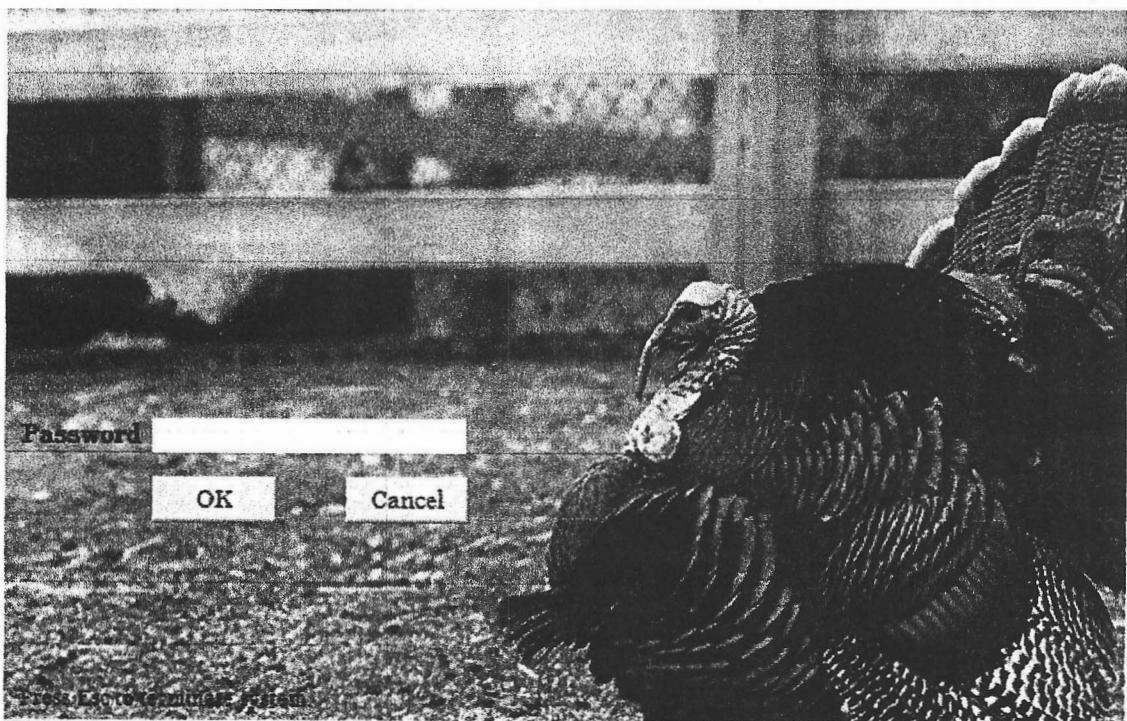
- Ali M.** David (2002), Nigeria, Poultry and Poultry products, Foreign Agricultural Service, Global Agricultural Network Report.
- Castle, E.N., M.H. Becker** and **F, J. Smith** (1972), Farm Business Management, The Decision Making Process, The Macmillan Company, New York, Chapter 3
- Duvich D.** Richard (2002), Computerized Farm Record Keeping, Department Of Agriculture, Environmental and Developmental Economics. The Ohio State University, Columbus.
- Microsoft Press** (1998), Learn Microsoft visual Basic Now, Microsoft Press.
- Steven Holzner** (1998), Visual Basic 6 Black Book, The Coriolis Group.
- Sturrock** (1971), Introduction to Farm Management Economics, The Pitman Press, Bath,
- Titlola** (1990), Poultry housing, The Macmillan Company.
- Ojo S. O.** (2003), Productivity and Technical Efficiency of Poultry Egg Production in Nigeria, Department of Agricultural Economics and Extension, Federal University of Technology, Akure, Nigeria, International Journal of Poultry Science

APPENDIX B: OUTPUT RESULT

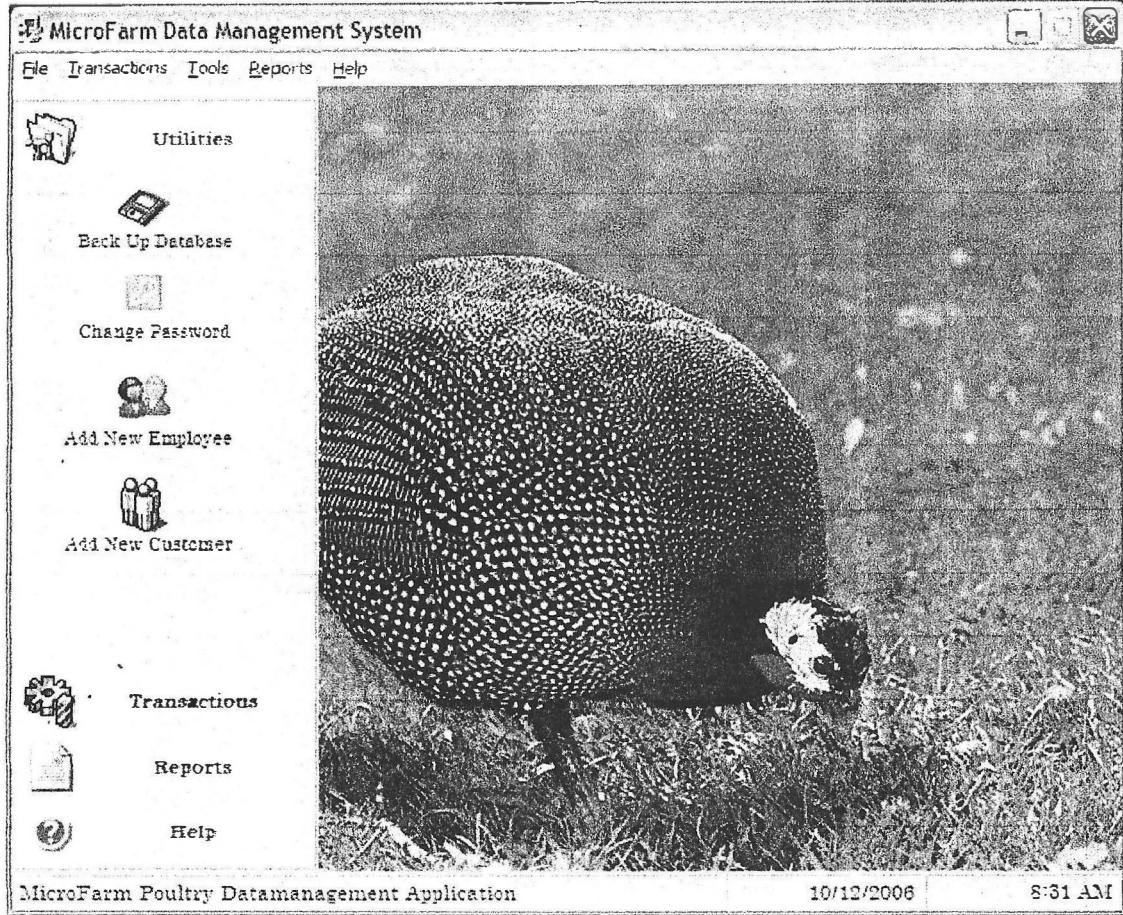
Fig 4.1: Splash Screen



Fig 4.2: Login Screen



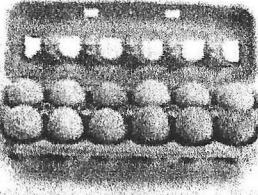
Main Screen



Activities Form

The screenshot shows a window titled "Activities". In the top right corner are standard window controls: a minimize button, a maximize button, and a close button. Below the title bar, the word "Date" is followed by a date input field showing "2010-01-01". The main area contains four data entry fields: "Description" (with an empty input field), "Type" (with an empty input field), "Number Affected" (with an empty input field), and "Unit" (with an empty input field). Below these fields is a large, empty rectangular area, likely a placeholder for a grid or list of items. At the bottom of the window are four buttons: "Add" (highlighted with a thick border), "Delete", "Cancel", and "Close".

Egg Production Form

 Egg Production

Date

Details	
Farm Unit 1	<input type="text"/>
Farm unit 2	<input type="text"/>
Total cracked	<input type="text"/>
Total produced (crates)	<input type="text"/>
Farm Unit 3	<input type="text"/>
Total produced	<input type="text"/>
Excess	<input type="text"/>

Remarks

Droppings production Form

Droppings Production

Date <input type="text" value="10/10/2008"/>	
Details	
Farm Unit 1	<input type="text"/>
Farm unit 2	<input type="text"/>
Farm Unit 3	<input type="text"/>
Total produced(bags)	<input type="text"/>
Remarks	<input type="text"/>
<input type="button" value="Add New"/> <input type="button" value="Clear"/> <input type="button" value="Save"/> <input type="button" value="Close"/>	

Employee's Information Form

Employee's Information



Employee Number:

Nick Name	[Text Box]		
First Name	[Text Box]		
Last Name	[Text Box]		
Street Address	[Text Box]		
State	[Text Box]		
Contact Number	[Text Box]		
Birthday	7/8/2008	Origin	[Text Box]
Sex	[Text Box]	[Text Box]	[Text Box]
Hire Date	7/8/2008	[Text Box]	[Text Box]

Remarks

Customer's Information Form

 Customer Information 

Customer Number

First Name

Last Name

Street Address

State

Contact Number

Sex

Remarks

Find Employee Form

Find Employee

Employee Num.	Nick Name	Name	Address
---------------	-----------	------	---------

Search by Employee Code
Employee Code |

Search by Name of Employee
Nick Name |
Last Name |
First Name |

Search Clear Cancel

Find Customer Form

Find Customer

Search by Customer Number Customer #	Result Customer Num. Name Address
---	--

Search by Name of Customer
Last Name |
First Name |

Search Clear < > Add New Record Close

Sales Form

 Sales 

Date Purchased:

Customer

 Customer

Customer Name:
Cust. Address:

Description:

Type: Unit Price: 0.00

Quantity: 0 Amount: 0.00

Add New Clear Save Close

Expenditure Form

Expenditure

Date Purchased: |

Description |

Quantity | 0 Unit Price | 0.00

Type | Amount | 0.00

Change Password Form

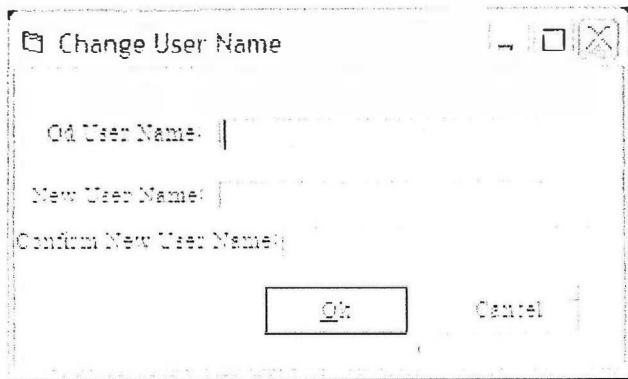
Change Password

Old Password: |

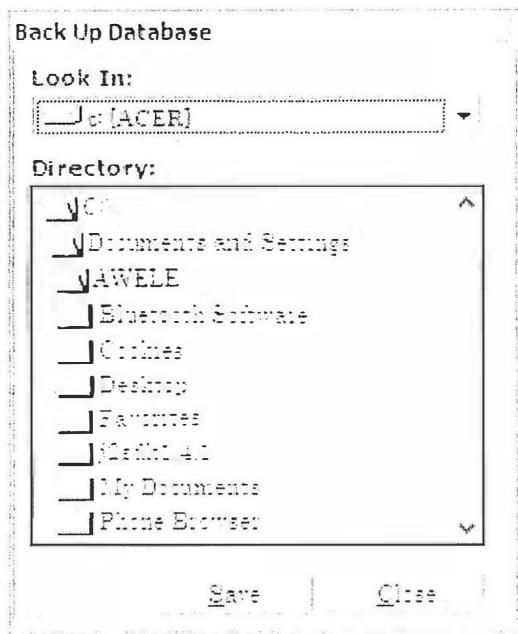
New Password: |

Confirm Password: |

Change User Name Form



Back-Up Database



Sales Report Form

 Sales Report X

Search Report Total Sales

Start Date Search | 0.00
End Date Clear |

OR #	Date	Customer Name	Item Description	Unit P.	Qty
------	------	---------------	------------------	---------	-----

Expenditure Report Form

Expenditure Report

P...	Date	Type	Item Description	Q...	Cost Per Unit
Total Purchases 0.00					

Search Report

Start Date: 7/15/2006 End Date: 7/15/2006

Search Clear

Navigation: < >

Help Form

About MICROFARM



MicroFarm
Version 1.0.0

This software is designed to meet the database requirements in Chicken-Rum Poultry Farm.

Warning! ...Copyright Any form of copying or duplicating of this software is strictly prohibited.
Intellectual property right

OK System Info...

Egg Production Report

Droppings Production Report

Droppings Production Report

Search Report Total Produced

Start Date Search End Date Clear

Date	Unit 1	Unit 2	Unit 3	Number of B...
				0

APPENDIX C:

DATABASE TABLES

ENTITIES

Employees

Customers

Sales

Expenditure

Password

Activities

Droppings Production

Egg Production

TABLES

Fig. 3.2 EMPLOYEES

Field Name	Data Type	Length
Employee Number	Text	12
Nick Name	Text	50
First Name	Text	50
Last Name	Text	50
Street Address	Memo	
State	Text	12
Contact Number	Number	10
Date of birth	Date/Time	
Place of Birth	Text	50
Sex	Text	10
Hire Date	Date/Time	
Image	image	

Fig. 3.3 CUSTOMERS

Field Name	Data Type	Length
Customer Name	Text	12
First Name	Text	12
Last Name	Text	50
Street Address	Memo	
State	Text	10
Sex	Text	10
Contact Number	Number	10

Fig: 3.4 SALES

Field Name	Data Type	Length
Sales Number	AutoNumber	
Date	Date/Time	
Customer Name	Text	50
Customer Address	Memo	
Item Description	Text	50
Type	Text	50
Quantity	Number	
Unit Price	Currency	
Amount	Currency	

Fig: 3.5 EXPENDITURE

Field Name	Data Type	Length
Purchase Code	AutoNumber	
Date	Date/Time	
Type	Text	50
Item Description	Text	50
Quantity	Number	
Cost	Currency	
Amount	Number	
Remark	Memo	

Fig: 3.6 EGG PRODUCTIONS

Field Name	Data Type	Length
Date	Date/Time	
Unit 1	Number	
Unit 2	Number	
Unit 3	Number	
Total Cracked	Number	

Total Produced	Number	
Excess	Number	
Remark	Memo	

Fig: 3.7 DROPPINGS PRODUCTION

Field Name	Data Type	Length
Date	Date/Time	
Unit 1	Number	
Unit 2	Number	
Unit 3	Number	
Number of Bags	Number	
Remark	Memo	

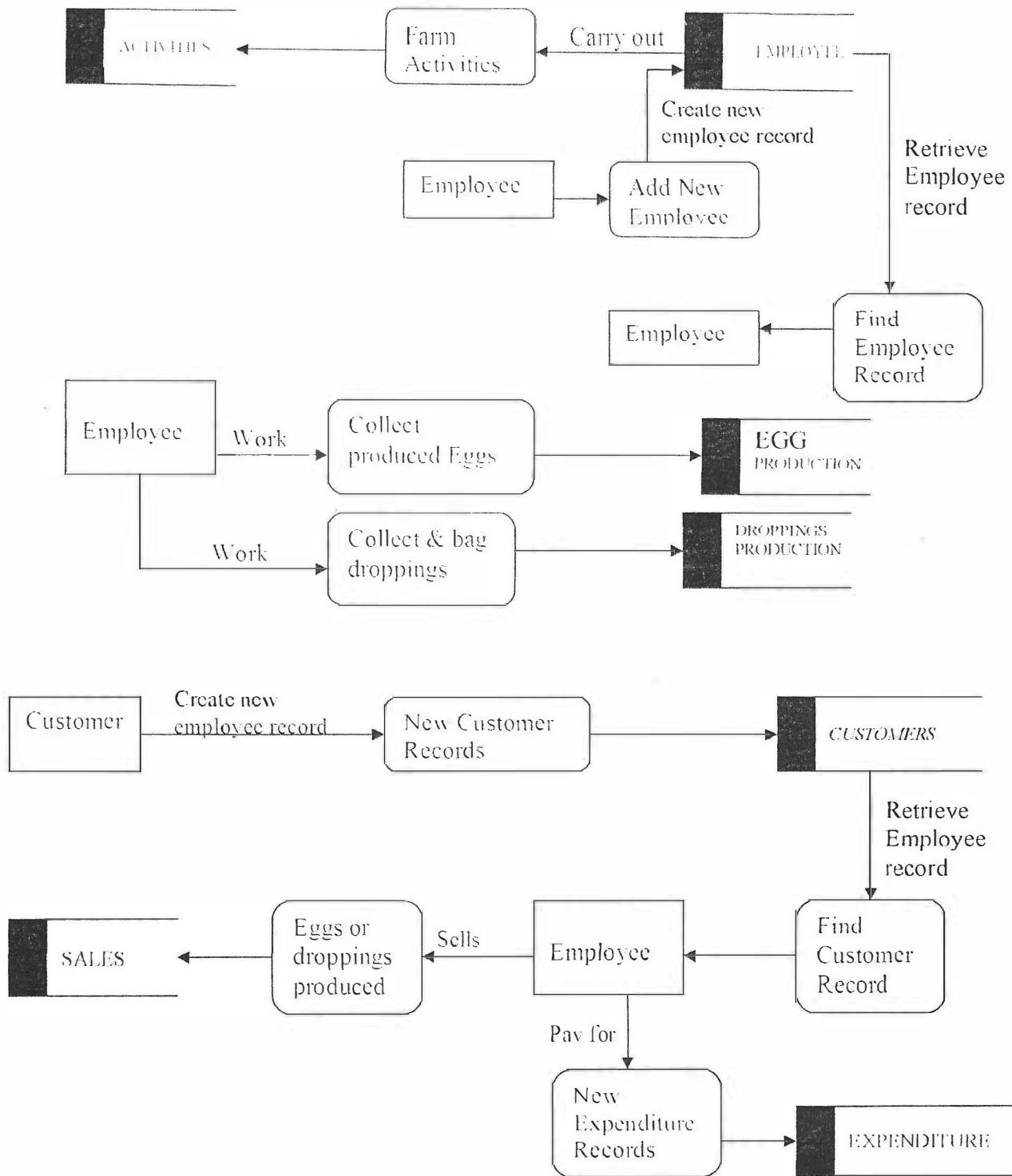
Fig: 3.8 LOGIN

Field Name	Data Type	Length
User Name	Text	50
Password	Text	50

Fig: 3.9 ACTIVITIES

Field Name	Data Type	Length
Date	Date/Time	
Description	Text	50
Type	Text	50
Unit	Number	10
Number Affected	Number	10
Remark	Memo	

Fig. 3.10

DATA FLOW DIAGRAM FOR PROCESSES

APPENDIX A: PROGRAM LISTING

Welcome Splash Screen

Option Explicit

```
Private Sub Form_KeyPress(KeyAscii As Integer)
    Unload Me
    Load frmMain
    Load frmLogin
    frmLogin.Show
End Sub
```

```
Private Sub Frame1_Click()
    Unload Me
    Load frmMain
    Load frmLogin
    frmLogin.Show
End Sub
```

```
Private Sub Timer1_Timer()
    Unload Me
    Load frmLogin
    frmLogin.Show
End Sub
```

Login

Option Explicit

```
Dim a As Integer
Dim counter
```

```
Private Sub cmdCancel_Click()
    Dim ans
    ans = MsgBox("Do you really want to terminate the system?", vbQuestion + vbYesNo)
    If ans = vbYes Then
        'Unload Me
        End
    End If
End Sub
```

```
Private Sub cmdOK_Click()
If (UCase(txt_Password) = UCase(txt_dbPassword)) And (UCase(Txt_UserName) =
UCase(Txt_dbUserName)) Then
    Unload Me
    Load frmMain
End If
```

```

Else
    MsgBox "Invalid Password/UserName.", vbExclamation
    Txt_UserName.SetFocus
    SendKeys "{home}" & "+{end}"
    counter = counter + 1
    If counter = 3 Then
        MsgBox "You have reached the maximum tries to enter your password." &
Chr(13) &_
        "System is terminating.", vbExclamation
    End
    End If
End If
End Sub

Private Sub Form_KeyPress(KeyAscii As Integer)
If KeyAscii = 27 Then
Dim ans
ans = MsgBox("Do you really want to terminate the system?", vbQuestion + vbYesNo)
If ans = vbYes Then
Unload Me
End If
End If
End Sub

Private Sub Form_Load()
a = 1
counter = 0

Data1.DatabaseName = App.Path & "\MICROFARM97.mdb"
Data1.RecordSource = "select * from Password"
Data1.Refresh
End Sub

Private Sub Timer1_Timer()
Label1.Caption = Label1.Caption & "."
a = a + 1
If a = 7 Then
    Timer1.Enabled = False
    Label1.Visible = False
    txt_Password.Visible = True
    cmdOK.Visible = True
    cmdCancel.Visible = True
    lbl_EnterPassword.Visible = True
    txt_Password.SetFocus
    lbl_UserName.Visible = True
    Txt_UserName.Visible = True

```

```
End If  
End Sub
```

```
Private Sub txt_Password_KeyPress(KeyAscii As Integer)  
If Key Ascii = 13 Then  
    If (UCase(txt_Password) = UCase(txt_dbPassword)) And (UCase(Txt_UserName) =  
UCase(Txt_dbUserName)) Then  
        Unload Me  
        Load frmMain  
        frmMain.Show  
        'unlock database here  
    Else  
        MsgBox "Invalid password/UserName.", vbExclamation  
        Txt_UserName.SetFocus  
        SendKeys "{home}" & "+{end}"  
        counter = counter + 1  
        If counter = 3 Then  
            MsgBox "You have reached the maximum tries to enter your password." &  
Chr(13) &  
            "System is terminating.", vbExclamation  
        End  
        End If  
    End If  
End If  
If Key Ascii = 27 Then  
Dim ans  
ans = MsgBox("Do you really want to terminate the system?", vbQuestion + vbYesNo)  
If ans = vbYes Then  
Unload Me  
End  
End If  
End If  
End Sub
```

Main form

```
Private Sub Form_Unload(Cancel As Integer)  
Cancel = 1  
Dim ans  
ans = MsgBox("Do you really want to terminate the system?", vbQuestion + vbYesNo)  
If ans = vbYes Then  
    'Unload Me  
    End  
    'frmSplash.Show  
Else  
    Exit Sub
```

```
End If  
End Sub
```

```
Private Sub Image1_Click(Index As Integer)  
Frame1(0).Visible = True  
Frame1(1).Visible = False  
Frame1(2).Visible = False  
Frame1(3).Visible = False  
End Sub
```

```
Private Sub Image12_Click()  
Load frmReportSales  
frmReportSales.Show  
End Sub
```

```
Private Sub Image15_Click()  
Load frmReportPurchase  
frmReportPurchase.Show  
End Sub
```

```
Private Sub Image16_Click()  
Load frmAbout  
frmAbout.Show  
End Sub
```

```
Private Sub Image2_Click(Index As Integer)  
Frame1(0).Visible = False  
Frame1(1).Visible = True  
Frame1(2).Visible = False  
Frame1(3).Visible = False  
End Sub
```

```
Private Sub Image3_Click(Index As Integer)  
Frame1(0).Visible = False  
Frame1(1).Visible = False  
Frame1(2).Visible = True  
Frame1(3).Visible = False  
End Sub
```

```
Private Sub Image4_Click(Index As Integer)  
Frame1(0).Visible = False  
Frame1(1).Visible = False  
Frame1(2).Visible = False  
Frame1(3).Visible = True  
End Sub
```

```
Private Sub Image5_Click()
```

```
Load frmPurchase
```

```
frmPurchase.Show
```

```
End Sub
```

```
Private Sub Image6_Click()
```

```
Load frmSales
```

```
frmSales.Show
```

```
End Sub
```

```
Private Sub img_AddNewCustomer_Click()
```

```
Load frmCustomer
```

```
frmCustomer.Show
```

```
End Sub
```

```
Private Sub img_AddNewEmployee_Click()
```

```
Load frmEmployee
```

```
frmEmployee.Show
```

```
End Sub
```

```
Private Sub img_BackUP_Click()
```

```
frmBackUp.Show
```

```
End Sub
```

```
Private Sub img_ChangePassword_Click()
```

```
frmChangePassword.Show
```

```
End Sub
```

```
Private Sub Label1_Click(Index As Integer)
```

```
Frame1(0).Visible = True
```

```
Frame1(1).Visible = False
```

```
Frame1(2).Visible = False
```

```
Frame1(3).Visible = False
```

```
End Sub
```

```
Private Sub Label2_Click(Index As Integer)
```

```
Frame1(0).Visible = False
```

```
Frame1(1).Visible = True
```

```
Frame1(2).Visible = False
```

```
Frame1(3).Visible = False
```

```
End Sub
```

```
Private Sub Label3_Click(Index As Integer)
```

```
Frame1(0).Visible = False
```

```
Frame1(1).Visible = False
```

```
Frame1(2).Visible = True
```

```
Frame1(3).Visible = False  
End Sub
```

```
Private Sub Label4_Click(Index As Integer)  
Frame1(0).Visible = False  
Frame1(1).Visible = False  
Frame1(2).Visible = False  
Frame1(3).Visible = True  
End Sub
```

```
Private Sub mnubackup_Click(Index As Integer)  
frmBackUp.Show  
End Sub
```

```
Private Sub mnuChange_Click(Index As Integer)  
frmChangePassword.Show  
End Sub
```

```
Private Sub mnuExit_Click(Index As Integer)  
Dim ans  
ans = MsgBox("Do you really want to terminate the system?", vbQuestion + vbYesNo)  
If ans = vbYes Then  
    'Unload Me  
    End  
    'frmSplash.Show  
Else  
    Exit Sub  
End If  
End Sub
```

```
Private Sub mnufileeggProd_Click(Index As Integer)  
Load frmEggProduction  
frmEggProduction.Show  
End Sub
```

```
Private Sub mnufileeggsales_Click(Index As Integer)  
Load frmSales  
frmSales.Show  
End Sub
```

```
Private Sub mnufileNewBirdssales_Click(Index As Integer)  
Load frmSales  
frmSales.Show  
End Sub
```

```
Private Sub mnufilenewdropprod_Click(Index As Integer)
```

```
Load FrmDroppingProd
FrmDroppingProd.Show
End Sub

Private Sub mnufilenewdropsales_Click(Index As Integer)
Load frmSales
frmSales.Show
End Sub

Private Sub mnufileegg_Click(Index As Integer)
Load frmEggProduction
frmEggProduction.Show
End Sub

Private Sub mnufileNewBirds_Click(Index As Integer)
Load frmActivities
frmActivities.Show
End Sub

Private Sub mnufilenewdrop_Click(Index As Integer)
Load frmDroppingsProduction
frmDroppingsProduction.Show
End Sub

Private Sub mnufindcustomer_Click(Index As Integer)
Load frmCustomerFind
frmCustomerFind.Show
End Sub

Private Sub mnuFindEmployee_Click(Index As Integer)
Load frmEmployeeFind
frmEmployeeFind.Show
End Sub

Private Sub mnuhelpproduct_Click(Index As Integer)
Load frmAbout
frmAbout.Show
End Sub

Private Sub mnunuProfileCustomer_Click(Index As Integer)
Load frmCustomer
frmCustomer.Show
End Sub

Private Sub mnunuProfileEmployee_Click(Index As Integer)
Load frmEmployee
```

```
frmEmployee.Show  
End Sub
```

```
Private Sub mnuPurchaseReport_Click(Index As Integer)  
Load frmReportPurchase  
frmReportPurchase.Show  
End Sub
```

```
Private Sub mnuSales_Click(Index As Integer)  
Load frmSales  
frmSales.Show  
End Sub
```

```
Private Sub mnuSalesreport_Click(Index As Integer)  
Load frmReportSales  
frmReportSales.Show  
End Sub
```

```
Private Sub MnutransExpen_Click(Index As Integer)  
Load frmPurchase  
frmPurchase.Show  
End Sub
```

```
Private Sub mnuEggProductionReport_Click(Index As Integer)  
Load frmReportEgg  
frmReportEgg.Show  
End Sub
```

```
Private Sub mnuDroppingproductionreport_Click(Index As Integer)  
Load frmReportDroppingProd  
frmReportDroppingProd.Show  
End Sub
```

Activities

```
Private Sub cmdAdd_Click()
```

```
'activate frames and controls  
DTPicker1.Enabled = True  
Frame1.Enabled = True  
Frame2.Enabled = True
```

```
cmdAdd.Enabled = False  
cmdSave.Enabled = True  
cmdCancel.Enabled = True
```

```
End Sub

Private Sub cmdCancel_Click()
clear_all_fields
End Sub

Private Sub cmdClose_Click()
Unload Me
End Sub

Private Sub cmdSave_Click()

If DTPicker1.Value = "" Then
MsgBox "Please Enter the Date.", vbOKOnly + vbCritical
DTPicker1.SetFocus
Exit Sub
End If

If Txt_Description.Text = "" Then
MsgBox "Please describe the activity you want to record.", vbOKOnly + vbCritical
Txt_Description.SetFocus
Exit Sub
End If

If Txt_NumberAffected.Text = "" Or IsNumeric(Txt_NumberAffected.Text) = False
Then
MsgBox "Please Enter a Valid Number", vbOKOnly + vbCritical
Txt_NumberAffected.SetFocus
Exit Sub
End If

If Txt_Type.Text = "" Then
MsgBox "Please enter the Type of activity", vbOKOnly + vbCritical
Txt_Type.SetFocus
Exit Sub
End If

If Txt_Unit.Text = "" Or IsNumeric(Txt_NumberAffected.Text) = False
Then
MsgBox "Please Enter a Valid Unit Number", vbOKOnly + vbCritical
Txt_Unit.SetFocus
Exit Sub
End If

If MsgBox("Confirm save New Activity Record.", vbQuestion + vbYesNo) = vbYes
Then
```

```
With Data1.Recordset
    .AddNew
        !Date = DTPicker1.Value
        !Description = Txt_Description.Text
        !Type = Txt_Type.Text
        !number_affected = Txt_NumberAffected.Text
        !unit = Txt_Unit.Text
        !Note = Txt_Remarks.Text
    .Update
End With
Data1.Refresh
MsgBox "New Activity Record Saved.", vbInformation
End If
```

```
'deactivate frames and controls
Frame1.Enabled = False
Frame2.Enabled = False

cmdSave.Enabled = False
cmdAdd.Enabled = True
cmdCancel.Enabled = False
```

```
clear_all_fields
End Sub
```

```
Private Sub Form_Load()
Data1.DatabaseName = App.Path & "\MICROFARM97.mdb"
Data1.RecordSource = "select * from Activities"
Data1.Refresh
End Sub
```

```
Private Sub clear_all_fields()

Txt_Description.Text = ""
Txt_NumberAffected.Text = ""
Txt_Remarks.Text = ""
Txt_Type.Text = ""
Txt_Unit.Text = ""
End Sub
```

Egg Production

```
Private Sub Cmd_Close_Click()
Unload Me
End Sub
```

```
Private Sub CmdAddNew_Click()
    clear_all_fields
```

```
' activate frames and controls
Frame1.Enabled = True
Frame2.Enabled = True
DTPicker1.Enabled = True
```

```
cmdClear.Enabled = True
cmdSave.Enabled = True
CmdAddNew.Enabled = False
End Sub
```

```
Private Sub cmdClear_Click()
    clear_all_fields
End Sub
```

```
Private Sub cmdSave_Click()
If MsgBox("Do you really want to save this Record?", vbQuestion + vbYesNo) = vbYes
Then
```

```
If Txt_totalProd = "" Then
    MsgBox "Please Enter the Quantity.", vbInformation + vbOKOnly
    Txt_totalProd.SetFocus
    Exit Sub
End If
```

```
If IsNumeric(Txt_totalProd) = False Or Val(Txt_totalProd) < 1 Then
    MsgBox "Please Enter a valid Number of Eggs.", vbInformation + vbOKOnly
    Txt_totalProd.SetFocus
    Exit Sub
End If
```

```
If IsNumeric(Txt_Unit1) = False Then
    MsgBox "Please Enter a valid Number of Eggs.", vbInformation + vbOKOnly
    Txt_Unit1.SetFocus
    Exit Sub
End If
```

```
If IsNumeric(Txt_Unit2) = False Then
    MsgBox "Please Enter a valid Number of Eggs.", vbInformation + vbOKOnly
    Txt_Unit2.SetFocus
    Exit Sub
End If
```

```
If IsNumeric(Txt_Unit3) = False Then
    }
```

```

MsgBox "Please Enter a valid Number of Eggs.", vbInformation + vbOKOnly
Txt_Unit3.SetFocus
Exit Sub
End If

If IsNumeric(Txt_Cracked) = False Then
    MsgBox "Please Enter a valid Number of Eggs.", vbInformation + vbOKOnly
    Txt_Unit3.SetFocus
    Exit Sub
End If

If MsgBox("Confirm save New Sales Record?", vbQuestion + vbYesNo) = vbYes
Then
    With Data1.Recordset
        .AddNew
        !Date = DTPicker1.Value
        !unit_1 = Txt_Unit1.Text
        !unit_2 = Txt_Unit2.Text
        !unit_3 = Txt_Unit3.Text
        !excess = TxtExcess.Text
        !total_cracked = Txt_Cracked.Text
        !total_produced = Txt_totalProd.Text
        !total_produced_in_crates = Txt_Crates.Text
        .Update
    End With
    Data1.Refresh
    MsgBox "New Egg Production Record Has been Saved!"
End If

Frame1.Enabled = False
Frame2.Enabled = False
DTPicker1.Enabled = False

cmdClear.Enabled = False
cmdSave.Enabled = False
CmdAddNew.Enabled = True

End If
End Sub

Private Sub Form_Load()
Data1.DatabaseName = App.Path & "\MICROFARM97.mdb"
Data1.RecordSource = "select * from EggProduction"
Data1.Refresh

Frame1.Enabled = False
Frame2.Enabled = False

```

```
DTPicker1.Enabled = False
```

```
cmdClear.Enabled = False  
cmdSave.Enabled = False  
CmdAddNew.Enabled = True
```

```
DTPicker1.Value = Format$(Now, "dd / mm / yyyy")  
End Sub
```

```
Private Sub Txt_Crates_Validate(Cancel As Boolean)  
Txt_Crates = Val(Txt_totalProd) \ 30  
End Sub
```

```
Private Sub Txt_totalProd_Validate(Cancel As Boolean)  
Txt_Crates = Val(Txt_totalProd) \ 30  
End Sub
```

```
Private Sub Txt_Unit1_Validate(Cancel As Boolean)  
Txt_totalProd.Text = Val(Txt_Unit1) + Val(Txt_Unit2) + Val(Txt_Unit3)  
End Sub
```

```
Private Sub Txt_Unit2_Validate(Cancel As Boolean)  
Txt_totalProd.Text = Val(Txt_Unit1) + Val(Txt_Unit2) + Val(Txt_Unit3)  
End Sub
```

```
Private Sub Txt_Unit3_Validate(Cancel As Boolean)  
Txt_totalProd.Text = Val(Txt_Unit1) + Val(Txt_Unit2) + Val(Txt_Unit3)  
End Sub
```

```
Private Sub clear_all_fields()  
Txt_Cracked.Text = ""  
Txt_Crates.Text = ""  
txt_Remark.Text = ""  
Txt_totalProd = ""  
Txt_Unit1 = ""  
Txt_Unit2.Text = ""  
Txt_Unit3.Text = ""  
TxtExcess.Text = ""  
End Sub
```

```
Private Sub TxtExcess_GotFocus()  
TxtExcess = Val(Txt_totalProd) Mod 30  
End Sub
```

Dropping Production

```
Private Sub Cmd_Close_Click()  
Unload Me  
End Sub
```

```
Private Sub CmdAddNew_Click()
clear_all_fields
```

```
' activate frames and controls
Frame1.Enabled = True
Frame2.Enabled = True
DTPicker1.Enabled = True
```

```
cmdClear.Enabled = True
cmdSave.Enabled = True
CmdAddNew.Enabled = False
End Sub
```

```
Private Sub cmdClear_Click()
    clear_all_fields
End Sub
```

```
Private Sub cmdSave_Click()
If MsgBox("Do you really want to Save this Record?", vbQuestion + vbYesNo) = vbYes
Then
```

```
If Txt_totalProd = "" Then
    MsgBox "Please Enter the Quantity.", vbInformation + vbOKOnly
    Txt_totalProd.SetFocus
    Exit Sub
End If
```

```
If IsNumeric(Txt_totalProd) = False Or Val(Txt_totalProd) < 1 Then
    MsgBox "Please Enter a valid Number of Eggs.", vbInformation + vbOKOnly
    Txt_totalProd.SetFocus
    Exit Sub
End If
```

```
If IsNumeric(Txt_Unit1) = False Then
    MsgBox "Please Enter a valid Number of Eggs.", vbInformation + vbOKOnly
    Txt_Unit1.SetFocus
    Exit Sub
End If
```

```
If IsNumeric(Txt_Unit2) = False Then
    MsgBox "Please Enter a valid Number of Eggs.", vbInformation + vbOKOnly
    Txt_Unit2.SetFocus
    Exit Sub
End If
```

```
If IsNumeric(Txt_Unit3) = False Then
```

```

MsgBox "Please Enter a valid Number of Eggs.", vbInformation + vbOKOnly
Txt_Unit3.SetFocus
Exit Sub
End If

If MsgBox("Confirm save New Droppings Production Record?", vbQuestion + vbYesNo) = vbYes Then
    With Data1.Recordset
        .AddNew
        !Date = DTPicker1.Value
        !unit_1 = Txt_Unit1.Text
        !unit_2 = Txt_Unit2.Text
        !unit_3 = Txt_Unit3.Text
        !number_of_bags = Txt_totalProd.Text
        !notes = txt_Remark.Text
        .Update
    End With
    Data1.Refresh
    MsgBox "New Dropping Production Record Has been Saved!"
    End If

    Frame1.Enabled = False
    Frame2.Enabled = False
    DTPicker1.Enabled = False

    cmdClear.Enabled = False
    cmdSave.Enabled = False
    CmdAddNew.Enabled = True
End If
End Sub

Private Sub Form_Load()
Data1.DatabaseName = App.Path & "\MICROFARM97.mdb"
Data1.RecordSource = "select * from DroppingsProduction"
Data1.Refresh

Frame1.Enabled = False
Frame2.Enabled = False
DTPicker1.Enabled = False

cmdClear.Enabled = False
cmdSave.Enabled = False
CmdAddNew.Enabled = True

DTPicker1.Value = Format$(Now, "dd / mm / yyyy")
End Sub

```

```

Private Sub Txt_Unit1_Validate(Cancel As Boolean)
Txt_totalProd.Text = Val(Txt_Unit1) + Val(Txt_Unit2) + Val(Txt_Unit3)
End Sub
Private Sub Txt_Unit2_Validate(Cancel As Boolean)
Txt_totalProd.Text = Val(Txt_Unit1) + Val(Txt_Unit2) + Val(Txt_Unit3)
End Sub
Private Sub Txt_Unit3_Validate(Cancel As Boolean)
Txt_totalProd.Text = Val(Txt_Unit1) + Val(Txt_Unit2) + Val(Txt_Unit3)
End Sub

```

```

Private Sub clear_all_fields()
txt_Remark.Text = ""
Txt_totalProd = ""
Txt_Unit1 = ""
Txt_Unit2.Text = ""
Txt_Unit3.Text = ""
End Sub

```

Employee Information

```

Option Explicit
Dim ans

```

```

Private Sub cmd_ClearPicture_Click()
CommDlg_Path.FileName = ""
Img_Emp.Picture = LoadPicture()
End Sub

```

```

Private Sub cmd_LoadPicture_Click()
With CommDlg_Path
.DialogTitle = "Search Employee picture"
.Filter = "Bitmap (*.bmp)|*.bmp|Jpeg (*.jpg)|*.jpg|Gif (*.gif)|*.gif>All Files (*.*)|*.*"
>ShowOpen
.FilterIndex = 1
.CancelError = True
Img_Emp.Picture = LoadPicture(.FileName)
End With
End Sub

```

```

Private Sub cmdAdd_Click()
Frame1.Enabled = True
Frame2.Enabled = True
txt_NickName.SetFocus
'controls

```

```
cmdAdd.Enabled = False  
cmdSave.Enabled = True  
cmdCancel.Enabled = True  
cmd_LoadPicture.Enabled = True  
cmd_ClearPicture.Enabled = True  
Employee_Number
```

```
End Sub
```

```
Private Sub cmdCancel_Click()  
'controls  
Frame1.Enabled = False  
Frame2.Enabled = False  
'controls  
cmdAdd.Enabled = True  
cmdSave.Enabled = False  
cmdCancel.Enabled = False  
cmd_LoadPicture.Enabled = False  
cmd_ClearPicture.Enabled = False  
clear_all_fields  
End Sub
```

```
Private Sub cmdClose_Click()  
Unload Me  
End Sub
```

```
Private Sub cmdSave_Click()  
If txt_NickName = "" Then  
    MsgBox "Cannot save if Nickname is empty.", vbExclamation, "Nickname is missing"  
    txt_NickName.SetFocus  
    Exit Sub  
End If
```

```
If txt_FirstName = "" Then  
    MsgBox "Cannot save if Firstname is empty.", vbExclamation, "Firstname is missing"  
    txt_FirstName.SetFocus  
    Exit Sub  
End If
```

```
If txt_Lastname = "" Then  
    MsgBox "Cannot save if Lastname is empty.", vbExclamation, "Lastname is missing"  
    txt_Lastname.SetFocus  
    Exit Sub  
End If
```

```
If txt_StreetAddress = "" Then
```

```
MsgBox "Cannot save if Street Address is empty.", vbExclamation, "Street Address is missing"
    txt_StreetAddress.SetFocus
    Exit Sub
End If

If txt_City = "" Then
    MsgBox "Cannot save if City is empty.", vbExclamation, "City is missing"
    txt_City.SetFocus
    Exit Sub
End If

If DTPicker1.Value = "" Then
    MsgBox "Cannot save if Birthday is empty.", vbExclamation, "Birth date is missing"
    DTPicker1.SetFocus
    Exit Sub
End If

If DTPicker2.Value = "" Then
    MsgBox "Cannot save if Date of hire is Empty.", vbExclamation, "date of hire is missing"
    DTPicker1.SetFocus
    Exit Sub
End If

'check if contact number is a valid numbers
If txt_ContactNumber.Text <> "" Then
    If IsNumeric(txt_ContactNumber.Text) = False Then
        MsgBox "Invalid contact number. Try again.", vbExclamation, "Error contact number"
        txt_ContactNumber.SetFocus
        SendKeys "{home}" & "{end}"
        Exit Sub
    End If
End If

If txt_birthplace.Text = "" Then
    MsgBox "Cannot save if Birthplace is empty.", vbExclamation, "Place of Origin is missing"
    txt_birthplace.SetFocus
    Exit Sub
End If

If combo_sex.Text = "" Then
    MsgBox "Cannot save if Gender is empty.", vbExclamation, "Gender is missing"
    combo_sex.SetFocus
```

```
Exit Sub  
End If
```

```
If MsgBox("Confirm add new Employee Record.", vbQuestion + vbYesNo) = vbYes  
Then
```

```
    With Data1.Recordset
```

```
        .AddNew
```

```
            !Emp_Nick_Name = UCase(txt_NickName.Text)  
            !Emp_FirstName = txt_FirstName.Text  
            !Emp_LastName = txt_Lastname.Text  
            !Emp_StreetAddress = txt_StreetAddress.Text  
            !Emp_State = txt_City.Text  
            !Emp_ContactNumber = Trim(txt_ContactNumber.Text)  
            !emp_birthday = DTPicker1.Value  
            !emp_hiredate = DTPicker2.Value  
            !emp_birthplace = txt_birthplace.Text  
            !emp_sex = combo_sex.Text  
            !emp_remark = txt_Remark.Text  
            !Emp_Number = lbl_EmployeeNumber.Caption  
            !emp_pic = CommDlg_Path.FileName
```

```
        .Update
```

```
    End With
```

```
    Data1.Refresh
```

```
    MsgBox "Employee information save.", vbInformation
```

```
End If
```

```
clear_all_fields
```

```
'controls
```

```
cmdAdd.Enabled = True  
cmdSave.Enabled = False  
cmdCancel.Enabled = True  
cmd_LoadPicture.Enabled = False  
cmd_ClearPicture.Enabled = False
```

```
End Sub
```

```
Private Sub clear_all_fields()
```

```
    txt_NickName.Text = ""  
    txt_FirstName.Text = ""  
    txt_Lastname.Text = ""  
    txt_StreetAddress.Text = ""  
    txt_City.Text = ""  
    txt_ContactNumber.Text = ""  
    txt_birthplace.Text = ""  
    combo_sex.Text = ""  
    txt_Remark.Text = ""
```

```
lbl_EmployeeNumber.Caption = "XXXXX"
cmd_ClearPicture_Click
End Sub
```

```
Private Sub Form_Load()
Data1.DatabaseName = App.Path & "\MICROFARM97.mdb"
Data1.RecordSource = "select * from Employees"
Data1.Refresh
combo_sex.Clear
combo_sex.AddItem "Male"
combo_sex.AddItem "Female"
End Sub
```

```
Private Sub Employee_Number()
Dim container1
Dim EmpCode
EmpCode = "MF-Emp-0000"
EmpCode = "MF-Emp-" & Int(Mid(EmpCode, 9)) + 1
Data1.RecordSource = "select * from Employees where Emp_Number = " & EmpCode
& ""
Data1.Refresh
While Data1.Recordset.RecordCount <> 0
    If Data1.Recordset.RecordCount <> 0 Then
        EmpCode = "MF-Emp-" & Int(Mid(EmpCode, 9)) + 1
    End If
    Data1.RecordSource = "select * from Employees where Emp_Number= " &
EmpCode & ""
    Data1.Refresh
Wend
lbl_EmployeeNumber = EmpCode
Data1.DatabaseName = App.Path & "\MICROFARM97.mdb"
Data1.RecordSource = "select * from Employees order by Emp_Number"
Data1.Refresh
End Sub
```

Customer Information

```
Option Explicit
Dim ans

Private Sub cmdAdd_Click()
clear_all_fields
'frame
fra_ci(0).Enabled = True
fra_ci(1).Enabled = True
```

```
txt_Firstname.SetFocus  
'controls  
cmdAdd.Enabled = False  
cmdSave.Enabled = True  
cmdCancel.Enabled = True  
Customer_Number  
End Sub
```

```
Private Sub cmdCancel_Click()  
'frame  
fra_ci(0).Enabled = False  
fra_ci(1).Enabled = False  
'controls  
cmdAdd.Enabled = True  
cmdSave.Enabled = False  
cmdCancel.Enabled = False  
clear_all_fields  
End Sub
```

```
Private Sub cmdClose_Click()  
Unload Me  
End Sub
```

```
Private Sub cmdSave_Click()  
If txt_Firstname = "" Then  
    MsgBox "Cannot save if Firstname is empty.", vbExclamation, "Firstname is missing"  
    txt_Firstname.SetFocus  
    Exit Sub  
End If  
If txt_Lastname = "" Then  
    MsgBox "Cannot save if Lastname is empty.", vbExclamation, "Lastname is missing"  
    txt_Lastname.SetFocus  
    Exit Sub  
End If  
If txt_StreetAddress = "" Then  
    MsgBox "Cannot save if Street Address is empty.", vbExclamation, "Street Address is missing"  
    txt_StreetAddress.SetFocus  
    Exit Sub  
End If  
If txt_City = "" Then  
    MsgBox "Cannot save if City or Town is empty.", vbExclamation, "City or Town is missing"  
    txt_City.SetFocus  
    Exit Sub  
End If
```

```

If txt_ContactNumber <> "" Then
    If IsNumeric(txt_ContactNumber) = False Then
        MsgBox "Invalid Contact number. Numbers only.", vbExclamation, "Invalid contact
number"
        txt_ContactNumber.SetFocus
        SendKeys "{home}" & "{end}"
        Exit Sub
    End If
End If
If combo_Sex = "" Then
    MsgBox "Cannot save if Gender is empty.", vbExclamation, "Gender is missing"
    combo_Sex.SetFocus
    Exit Sub
End If

If MsgBox("Confirm save new customer record.", vbQuestion + vbYesNo) = vbYes
Then
    With Data1.Recordset
        .AddNew
        !Cust_Firstname = txt_Firstname.Text
        !Cust_Lastname = txt_Lastname.Text
        !Cust_StreetAddress = txt_StreetAddress.Text
        !Cust_State = txt_City.Text
        !Cust_ContactNumber = Trim(txt_ContactNumber.Text)
        !Cust_sex = combo_Sex.Text
        !Cust_remarks = txt_Remarks.Text
        !Cust_Number = LBL_CustomerNumber.Caption
        .Update
    End With
    Data1.Refresh
    MsgBox "New customer record saved.", vbInformation
End If

'frame
fra_ci(0).Enabled = False
fra_ci(1).Enabled = False
'controls
cmdAdd.Enabled = True
cmdSave.Enabled = False
cmdCancel.Enabled = False
clear_all_fields
End Sub

Private Sub Form_Load()
Data1.DatabaseName = App.Path & "\MICROFARM97.mdb"
Data1.RecordSource = "select * from CustomerS order by Cust_Number"

```

```
Data1.Refresh
```

```
combo_Sex.Clear  
combo_Sex.AddItem "Male"  
combo_Sex.AddItem "Female"  
End Sub
```

```
Private Sub clear_all_fields()  
txt_Firstname.Text = ""  
txt_Lastname.Text = ""  
txt_StreetAddress.Text = ""  
txt_City.Text = ""  
txt_ContactNumber.Text = ""  
combo_Sex.Text = ""  
txt_Remarks.Text = ""  
LBL_CustomerNumber.Caption = "XXXXX"  
End Sub  
Private Sub Customer_Number()  
Dim container1  
Dim CustCode  
CustCode = "C-0"  
CustCode = "C-" & Int(Mid(CustCode, 3)) + 1  
Data1.RecordSource = "select * from Customers where Cust_Number = " & CustCode  
& """  
Data1.Refresh  
While Data1.Recordset.RecordCount <> 0  
    If Data1.Recordset.RecordCount <> 0 Then  
        CustCode = "C-" & Int(Mid(CustCode, 3)) + 1  
    End If  
    Data1.RecordSource = "select * from Customers where Cust_Number= " &  
CustCode & """  
    Data1.Refresh  
Wend  
LBL_CustomerNumber = CustCode  
Data1.DatabaseName = App.Path & "\MICROFARM97.mdb"  
Data1.RecordSource = "select * from Customers order by Cust_Number"  
Data1.Refresh  
End Sub
```

Find Employee

```
Option Explicit  
Dim l_results As ListItem
```

```
Private Sub cmd_cancel_Click()  
cmd_Clear_Click
```

```
Unload Me  
End Sub
```

```
Private Sub cmd_Clear_Click()  
Timer1.Enabled = False  
lvw_recordsfound.ListItems.Clear  
txt_employeecode.Text = ""  
txt_NickName.Text = ""  
txt_firstname.Text = ""  
txt_lastname.Text = ""  
cmd_search.Enabled = False  
Data1.DatabaseName = App.Path & "\MICROFARM97.mdb"  
Data1.RecordSource = "select * from Employees"  
Data1.Refresh  
cmd_search.Enabled = True  
End Sub
```

```
Private Sub cmd_Search_Click()
```

```
If txt_employeecode.Text = "" And txt_NickName.Text = "" And txt_firstname.Text = ""  
And txt_lastname.Text = "" Then  
    MsgBox "Sorry, cannot search right now. Try to enter a value.", vbExclamation  
    cmd_search.Enabled = True  
    Exit Sub  
Else  
    cmd_search.Enabled = False  
End If  
lvw_recordsfound.ListItems.Clear
```

```
With Data1
```

```
'employee code  
If txt_employeecode.Text <> "" Then  
    'no record found  
    .RecordSource = "select * from Employees where Emp_Number ='" &  
    Trim(txt_employeecode.Text) & "'"  
    .Refresh  
    If Recordset.RecordCount = 0 Then  
        MsgBox "No record found with employee code: " &  
        Trim(txt_employeecode.Text) & ". "  
        .DatabaseName = App.Path & "\MICROFARM97.mdb"  
        .RecordSource = "select * from Employees"  
        .Refresh  
        Exit Sub  
    Else  
        'record/s found
```

```

'cmd_search.Enabled = False
Populate_lvw_recordsfound
Timer1.Enabled = True
.DatabaseName = App.Path & "\MICROFARM97.mdb"
.RecordSource = "select * from Employees"
.Refresh

End If
End If

'nick name
If txt_NickName.Text <> "" Then
  'no record found
  .RecordSource = "select * from Employees where Emp_Nick_Name = "" &
Trim(txt_NickName.Text) & """
  .Refresh
  If .Recordset.RecordCount = 0 Then
    MsgBox "No record found with Nickname: " & UCase(txt_NickName.Text) & "."
    .DatabaseName = App.Path & "\MICROFARM97.mdb"
    .RecordSource = "select * from Employees"
    .Refresh
    Exit Sub
  Else
    'record/s found
    'cmd_search.Enabled = False
    Populate_lvw_recordsfound
    Timer1.Enabled = True
    .DatabaseName = App.Path & "\MICROFARM97.mdb"
    .RecordSource = "select * from Employees"
    .Refresh

  End If
End If

'first name
If txt_firstname.Text <> "" Then
  'no record found
  .RecordSource = "select * from Employees where Emp_FirstName = "" &
Trim(txt_firstname.Text) & """
  .Refresh
  If .Recordset.RecordCount = 0 Then
    MsgBox "No record found with Firstname: " & UCase(txt_firstname.Text) & "."
    .DatabaseName = App.Path & "\MICROFARM97.mdb"
    .RecordSource = "select * from Employees"
    .Refresh
    Exit Sub
  End If
End If

```

```

Else
    'record/s found
    'cmd_search.Enabled = False
    Populate_lvw_recordsfound
    Timer1.Enabled = True
    .DatabaseName = App.Path & "\MICROFARM97.mdb"
    .RecordSource = "select * from Employees"
    .Refresh

    End If
End If

'last name
If txt_lastname.Text <> "" Then
    'no record found
    .RecordSource = "select * from Employees where Emp_LastName = '" &
Trim(txt_lastname.Text) & "'"
    .Refresh

    If .Recordset.RecordCount = 0 Then
        MsgBox "No record found with Lastname: " & UCase(txt_lastname.Text) & "."
        .DatabaseName = App.Path & "\MICROFARM97.mdb"
        .RecordSource = "select * from Employees"
        .Refresh
        Exit Sub

    Else
        'record/s found
        'cmd_search.Enabled = False
        Populate_lvw_recordsfound
        Timer1.Enabled = True
        .DatabaseName = App.Path & "\MICROFARM97.mdb"
        .RecordSource = "select * from Employees"
        .Refresh

    End If
End If

End With
End Sub

Private Sub Populate_lvw_recordsfound()
Dim rec_count, X
On Err GoTo hell

    'Timer1.Enabled = True
    With Data1.Recordset

```

```
.MoveLast  
rec_count = .RecordCount  
.MoveFirst  
For X = 1 To rec_count  
Set l_results = lvw_recordsfound.ListItems.Add(X, , !Emp_Number)  
    l_results.SubItems(1) = !Emp_Nick_Name  
    l_results.SubItems(2) = !Emp_FirstName & " " & !Emp_LastName  
    l_results.SubItems(3) = !Emp_StreetAddress & ", " & !Emp_State  
    l_results.SubItems(4) = !Emp_ContactNumber  
  
.MoveNext  
Next  
    End With  
Exit Sub  
End Sub
```

```
Private Sub cmd_select_Click()  
MsgBox lvw_recordsfound.SelectedItem & " " & l_results.SubItems(1)  
End Sub
```

```
Private Sub Form_Load()  
Data1.DatabaseName = App.Path & "\MICROFARM97.mdb"  
Data1.RecordSource = "select * from Employees"  
Data1.Refresh  
  
End Sub
```

```
Private Sub lvw_recordsfound_Click()  
'MsgBox lvw_recordsfound.SelectedItem  
End Sub
```

```
Private Sub lvw_recordsfound_ColumnClick(ByVal ColumnHeader As  
MSComctlLib.ColumnHeader)  
If lvw_recordsfound.SortOrder = lvwAscending Then  
    lvw_recordsfound.SortOrder = lvwDescending  
Else  
    lvw_recordsfound.SortOrder = lvwAscending  
End If  
End Sub
```

```
Private Sub Timer1_Timer()  
  
If lvw_recordsfound.SelectedItem <> "" Or lvw_recordsfound.SelectedItem <> Null  
Then  
    With Data1.Recordset
```

```
Data1.RecordSource = "select * from Employees where Emp_Number = "" &
lvw_recordsfound.SelectedItem & """
Data1.Refresh
If .RecordCount <> 0 Then
    On Error Resume Next
    Img_Emp.Picture = LoadPicture(!emp_pic)
    If Img_Emp.Picture = 0 Then
        Img_Emp.Picture = LoadPicture()
    Else
        If !emp_pic <> "" Or !emp_pic <> Null Then
            Img_Emp.Picture = LoadPicture(!emp_pic)
        Else
            Img_Emp.Picture = LoadPicture()
        End If
    End If
Else
    Img_Emp.Picture = LoadPicture()
End If
End With
Else
    Exit Sub
End If
End Sub
```

```
Private Sub txt_employeocode_Change()
'If txt_employeocode.Text <> "" Then
    ' txt_NickName.Text = ""
    ' txt_Firstname.Text = ""
    ' txt_Lastname.Text = ""
    cmd_search.Enabled = True
'Else
    ' cmd_search.Enabled = False
'End If
End Sub
```

```
Private Sub txt_employeocode_GotFocus()
txt_employeocode.Text = "MFEmp-"
txt_employeocode.SetFocus
```

```
End Sub
```

```
Private Sub txt_Firstname_Change()
'If txt_Firstname.Text <> "" Then
    ' txt_employeocode.Text = ""
    ' txt_NickName.Text = ""
    ' txt_Lastname.Text = ""
```

```
' cmd_search.Enabled = True  
'Else  
' cmd_search.Enabled = False  
'End If  
End Sub
```

```
Private Sub txt_Lastname_Change()  
'If txt_Lastname.Text <> "" Then  
' txt_employeeecode.Text = ""  
' txt_NickName.Text = ""  
' txt_Firstname.Text = ""  
' cmd_search.Enabled = True  
'Else  
' cmd_search.Enabled = False  
'End If  
End Sub
```

```
Private Sub txt_NickName_Change()  
'If txt_NickName.Text <> "" Then  
' txt_employeeecode.Text = ""  
' txt_Firstname.Text = ""  
' txt_Lastname.Text = ""  
' cmd_search.Enabled = True  
'Else  
' cmd_search.Enabled = False  
'End If  
End Sub
```

Find Customer Information

```
Option Explicit  
Dim l_results As ListItem
```

```
Private Sub cmd_AddNew_Click()  
frmCustomer.Show vbModal  
End Sub
```

```
Private Sub cmd_cancel_Click()  
Unload Me  
End Sub
```

```
Private Sub cmd_Clear_Click()  
txt_CustomerNumber.Text = ""  
txt_Firstname.Text = ""  
txt_Lastname.Text = ""  
lvw_recordsfound.ListItems.Clear
```

```

cmd_search.Enabled = False
Data1.DatabaseName = App.Path & "\MICROFARM97.mdb"
Data1.RecordSource = "select * from Customers"
Data1.Refresh

End Sub

Private Sub cmd_Search_Click()
If txt_CustomerNumber.Text = "" And txt_Firstname.Text = "" And txt_Lastname.Text =
"" Then
    MsgBox "Sorry, cannot search right now. Try to enter a value.", vbExclamation
    Exit Sub
End If

With Data1
    lvw_recordsfound.ListItems.Clear
    'customer number
    If txt_CustomerNumber.Text <> "" Then
        'no record found
        .RecordSource = "select * from Customers where Cust_Number ='" &
        Trim(txt_CustomerNumber.Text) & "'"
        .Refresh
        If .Recordset.RecordCount = 0 Then
            MsgBox "No record found with customer number: " &
            Trim(txt_CustomerNumber.Text) & "."
            .DatabaseName = App.Path & "\MICROFARM97.mdb"
            .RecordSource = "select * from Customers"
            .Refresh
            cmd_select.Enabled = False
            Exit Sub
        Else
            'record/s found
            cmd_select.Enabled = True
            Populate_lvw_recordsfound
            .DatabaseName = App.Path & "\MICROFARM97.mdb"
            .RecordSource = "select * from Customers"
            .Refresh
            Exit Sub
        End If
    End If

    lvw_recordsfound.ListItems.Clear
    'first name
    If txt_Firstname.Text <> "" Then
        'no record found

```

```
.RecordSource = "select * from customers where Cust_firstname ='" &
txt_Firstname.Text & "'"
.Refresh
If .Recordset.RecordCount = 0 Then
    MsgBox "No record found with firstname: " & Trim(txt_Firstname.Text) & "."
    .DatabaseName = App.Path & "\MICROFARM97.mdb"
    .RecordSource = "select * from Customers"
    .Refresh
    cmd_select.Enabled = False
    Exit Sub
Else
    'record/s found
    cmd_select.Enabled = True
    Populate_lvw_recordsfound
    .DatabaseName = App.Path & "\MICROFARM97.mdb"
    .RecordSource = "select * from Customers"
    .Refresh
    Exit Sub
End If
End If
```

```
lvw_recordsfound.ListItems.Clear
'lastname name
If txt_Lastname.Text <> "" Then
    'no record found
    .RecordSource = "select * from customers where Cust_lastname ='" &
Trim(txt_Lastname.Text) & "'"
    .Refresh
    If .Recordset.RecordCount = 0 Then
        MsgBox "No record found with lastname: " & Trim(txt_Lastname.Text) & "."
        .DatabaseName = App.Path & "\MICROFARM97.mdb"
        .RecordSource = "select * from Customers"
        .Refresh
        cmd_select.Enabled = False
        Exit Sub
    Else
        'record/s found
        cmd_select.Enabled = True
        Populate_lvw_recordsfound
        .DatabaseName = App.Path & "\MICROFARM97.mdb"
        .RecordSource = "select * from Customers"
        .Refresh
        Exit Sub
    End If
End If
```

```
End With
```

```
End Sub
```

```
Private Sub cmd_select_Click()
frmSales.txt_CustomerName.Text = l_results.SubItems(1)
frmSales.txt_CustomerAddress.Text = l_results.SubItems(2)
Unload Me
End Sub
```

```
Private Sub Form_Load()
Data1.DatabaseName = App.Path & "\MICROFARM97.mdb"
Data1.RecordSource = "select * from Customers"
Data1.Refresh
End Sub
```

```
'pre defined functions
```

```
Private Sub Populate_lvw_recordsfound()
Dim rec_count, X
With Data1.Recordset
    .MoveLast
    rec_count = .RecordCount
    .MoveFirst
    For X = 1 To rec_count
        Set l_results = lvw_recordsfound.ListItems.Add(X, , !Cust_Number)
        l_results.SubItems(1) = !Cust_Firstname & " " & !Cust_Lastname
        l_results.SubItems(2) = !Cust_StreetAddress & ", " & !Cust_State
        l_results.SubItems(3) = !Cust_ContactNumber
    .MoveNext
    Next
End With
End Sub
```

```
Private Sub lvw_recordsfound_ColumnClick(ByVal ColumnHeader As
MSComctlLib.ColumnHeader)
    If lvw_recordsfound.SortOrder = lvwAscending Then
        lvw_recordsfound.SortOrder = lvwDescending
    Else
        lvw_recordsfound.SortOrder = lvwAscending
    End If
End Sub
```

```
Private Sub txt_CustomerNumber_Change()
If txt_CustomerNumber.Text <> "" Then
```

```
txt_Firstname.Text = ""  
txt_Lastname.Text = ""  
cmd_search.Enabled = True  
Else  
    cmd_search.Enabled = False  
End If  
End Sub
```

```
Private Sub txt_CustomerNumber_GotFocus()  
txt_CustomerNumber.Text = "C-"  
End Sub
```

```
Private Sub txt_Firstname_Change()  
If txt_Firstname.Text <> "" Then  
    txt_Lastname.Text = ""  
    txt_CustomerNumber.Text = ""  
    cmd_search.Enabled = True  
Else  
    cmd_search.Enabled = False  
End If  
End Sub
```

```
Private Sub txt_Lastname_Change()  
If txt_Lastname.Text <> "" Then  
    txt_Firstname.Text = ""  
    txt_CustomerNumber.Text = ""  
    cmd_search.Enabled = True  
Else  
    cmd_search.Enabled = False  
End If  
  
End Sub
```

Sales Form

```
Private Sub Cmd_Close_Click()  
Unload Me  
End Sub
```

```
Private Sub cmd_find_Click()  
frmCustomerFind.Show  
End Sub
```

```
Private Sub CmdAddNew_Click()  
'activate frames and controls  
DTPicker1.Enabled = True
```

```
Frame1(2).Enabled = True  
Frame1(3).Enabled = True
```

```
CmdAddNew.Enabled = False  
cmdSave.Enabled = True  
cmdClear.Enabled = True  
End Sub
```

```
Private Sub cmdClear_Click()  
clear_all_fields  
End Sub
```

```
Private Sub cmdSave_Click()  
If txt_CustomerName.Text = "" Then  
    MsgBox "Please enter customer's name.", vbExclamation, "Customer's name is missing"  
    txt_CustomerName.SetFocus  
    Exit Sub  
End If  
  
If txt_CustomerAddress.Text = "" Then  
    MsgBox "Please enter customer's address.", vbExclamation, "Customer's address is missing"  
    txt_CustomerAddress.SetFocus  
    Exit Sub  
End If  
  
If Combo1 = "" Then  
    MsgBox "Please choose a product to be sold.", vbExclamation, "Product name is missing"  
    Combo1.SetFocus  
    Exit Sub  
End If
```

```
If IsNumeric(txt_Quantity) = False Then  
    MsgBox "Invalid quantity. Please enter a valid quantity. Numbers only.",  
vbExclamation, "Invalid quantity"  
    txt_Quantity.SetFocus  
    SendKeys "{home}" & "+{end}"  
    Exit Sub  
End If
```

```
If Val(txt_Quantity) < 1 Then  
    MsgBox "Please enter a valid quantity.", vbExclamation, "Invalid quantity"  
    txt_Quantity.SetFocus  
    SendKeys "{home}" & "+{end}"
```

```
Exit Sub  
End If
```

```
If Val(lbl_Amount) < 1 Then
```

```
    MsgBox "Invalid amount. Please choose a product.", vbExclamation, "Invalid amount"
```

```
    SendKeys "{home}" & "+{end}"
```

```
    Exit Sub
```

```
End If
```

```
If IsNumeric(txt_Quantity) = False Or txt_Quantity = "" Then
```

```
    MsgBox "Invalid quantity. Please enter a valid quantity. Numbers only.",  
vbExclamation, "Invalid quantity"
```

```
    txt_Quantity.SetFocus
```

```
    SendKeys "{home}" & "+{end}"
```

```
    Exit Sub
```

```
End If
```

```
If Val(txt_Quantity) < 1 Then
```

```
    MsgBox "Please enter a valid quantity.", vbExclamation, "Invalid quantity"
```

```
    txt_Quantity.SetFocus
```

```
    SendKeys "{home}" & "+{end}"
```

```
    Exit Sub
```

```
End If
```

```
If MsgBox("Do you really want to sell the product?", vbQuestion + vbYesNo) = vbYes
```

```
Then
```

```
With Data1.Recordset
```

```
    .AddNew
```

```
    !Date_Sold = DTPicker1.Value
```

```
    !Customer_Name = txt_CustomerName.Text
```

```
    !Customer_Address = txt_CustomerAddress.Text
```

```
    !Type = Cmb_type.Text
```

```
    !Item_Description = Combo1.Text
```

```
    !Qty = txt_Quantity.Text
```

```
    !Unit_Price = txt_cOst.Text
```

```
    !Amount = lbl_Amount
```

```
    .Update
```

```
    End With
```

```
    Data1.Refresh
```

```
    MsgBox "Product/s has been sold.", vbInformation, "Purchased"
```

```
'deactivate frames and controls
```

```
DTPicker1.Enabled = False
```

```
Frame1(2).Enabled = False
```

```
Frame1(3).Enabled = False
```

```
cmdSave.Enabled = False  
cmdClear.Enabled = False  
CmdAddNew.Enabled = True  
  
clear_all_fields  
Exit Sub  
End If  
End Sub
```

```
Private Sub Combo1_Validate(Cancel As Boolean)  
If Combo1.Text = "Eggs" Then  
    Cmb_type.Clear  
    Cmb_type.AddItem "XLarge"  
    Cmb_type.AddItem "Large"  
    Cmb_type.AddItem "Medium"  
    Cmb_type.AddItem "Midi"  
    Cmb_type.AddItem "Small"  
  
ElseIf Combo1.Text = "Birds" Then  
    Cmb_type.Clear  
    Cmb_type.AddItem "Layers"  
    Cmb_type.AddItem "Broilers"  
    Cmb_type.AddItem "Hens"  
    Cmb_type.AddItem "Turkey"  
    Cmb_type.AddItem "Guinea Fowl"  
  
ElseIf Combo1.Text = "Droppings" Then  
    Cmb_type.Clear  
    Cmb_type.AddItem "Dried"  
    Cmb_type.AddItem "Fresh"  
End If  
  
End Sub
```

```
Private Sub Form_Load()  
  
sales  
Data1.DatabaseName = App.Path & "\MICROFARM97.mdb"  
Data1.RecordSource = "select * from Sales"  
Data1.Refresh  
  
End Sub
```

```
Private Sub lbl_Amount_Click()  
lbl_Amount.Caption = Val(txt_cOst.Text) * Val(txt_Quantity.Text)  
End Sub
```

```
Private Sub txt_cOst_Change()
lbl_Amount.Caption = Val(txt_Quantity.Text) * Val(Format(txt_cOst,
"#####.####.00"))
lbl_Amount.Caption = Format(lbl_Amount, "###,###,###,##.00")
End Sub
```

```
Private Sub txt_cOst_LostFocus()
lbl_Amount.Caption = Val(txt_Quantity.Text) * Val(Format(txt_cOst,
"#####.####.00"))
lbl_Amount.Caption = Format(lbl_Amount, "###,###,###,##.00")
End Sub
```

```
Private Sub txt_Quantity_Change()
lbl_Amount.Caption = Val(txt_Quantity.Text) * Val(Format(txt_cOst,
"#####.####.00"))
lbl_Amount.Caption = Format(lbl_Amount, "###,###,###,##.00")
End Sub
```

```
Private Sub clear_all_fields()
txt_ORNumber.Text = ""
txt_CustomerName.Text = ""
txt_CustomerAddress.Text = ""
Combo1.Clear
Cmb_type.Clear
txt_Quantity.Text = "0"
txt_cOst.Text = "0.00"
lbl_Amount.Caption = "0.00"
End Sub
```

```
Private Sub txt_Quantity_LostFocus()
lbl_Amount = Val(txt_Quantity) * Val(txt_cOst)
End Sub
```

Expenditure Form

```
Option Explicit
Dim rec_count, i, X
```

```
Private Sub Cmd_Close_Click()
Unload Me
End Sub
```

```
Private Sub CmdAddNew_Click()
'activate frames and controls
DTPicker1.Enabled = True
```

```
Frame1(2).Enabled = True  
Frame2.Enabled = True  
  
CmdAddNew.Enabled = False  
cmdSave.Enabled = True  
cmdClear.Enabled = True  
clear_all_fields  
End Sub
```

```
Private Sub cmdClear_Click()  
    txt_Quantity = ""  
    Txt_ItemDescription = ""  
    txt_cOst = ""  
    lbl_Amount = ""  
    TxtType = ""  
End Sub
```

```
Private Sub cmdSave_Click()  
If MsgBox("Do you really want to buy the product?", vbQuestion + vbYesNo) = vbYes  
Then
```

```
If txt_Quantity = "" Then  
    MsgBox "Please Enter the Quantity.", vbInformation + vbOKOnly  
    txt_Quantity.SetFocus  
    Exit Sub  
End If
```

```
If txt_cOst = "" Then  
    MsgBox "Please Enter the Unit Price.", vbInformation + vbOKOnly  
    txt_cOst.SetFocus  
    Exit Sub  
End If
```

```
If IsNumeric(txt_cOst) = False Or Val(txt_cOst) < 1 Then  
    MsgBox "Please Enter a valid Unit Price.", vbInformation + vbOKOnly  
    txt_cOst.SetFocus  
    Exit Sub  
End If
```

```
If MsgBox("Confirm save New Expenditure Record?", vbQuestion + vbYesNo) =  
vbYes Then  
    With Data1.Recordset  
        .AddNew  
        !Date_purchased = DTPicker1.Value  
        !Item_Description = Txt_ItemDescription.Text  
        !Type = TxtType.Text  
        !Qty = txt_Quantity.Text
```

```

!cost = txt_cOst.Text
!Amount = lbl_Amount.Caption
!remarks = Txt_Remarks.Text
.Update
End With
Data1.Refresh
MsgBox "The New record has been saved", vbInformation + vbOKOnly
Txt_ItemDescription = ""
txt_Quantity = ""
txt_cOst = ""
lbl_Amount = ""
TxtType = ""
Txt_Remarks = ""

'deactivate frames and control
DTPicker1.Enabled = False
Frame1(2).Enabled = False
Frame2.Enabled = False

CmdAddNew.Enabled = True
cmdSave.Enabled = True
cmdClear.Enabled = False
End If
End If
End Sub

```

```

Private Sub Form_Load()
Data1.DatabaseName = App.Path & "\MICROFARM97.mdb"
Data1.RecordSource = "select * from Expenditure"
Data1.Refresh
End Sub

```

```

Private Sub txt_cOst_Change()
If IsNumeric(txt_cOst.Text) = False Then
    MsgBox "Invalid cost, enter a valid cost of product (Numbers only).", vbExclamation,
"Invalid cost amount"
    SendKeys "{home}" & "+{end}"
Else
    lbl_Amount.Caption = Val(txt_Quantity.Text) * Val(Format(txt_cOst,
"#####.##"))
    lbl_Amount.Caption = Format(lbl_Amount, "###,###,###,###.##")
End If
End Sub

```

```

Private Sub txt_Quantity_Change()

```

```
lbl_Amount.Caption = Val(txt_Quantity.Text) * Val(Format(txt_cOst,  
"#####.##"))  
lbl_Amount.Caption = Format(lbl_Amount, "###,###,###,###.00")  
End Sub
```

```
Private Sub txt_Quantity_GotFocus()  
SendKeys "{home}" & "+{end}"  
End Sub
```

```
Private Sub clear_all_fields()  
Txt_ItemDescription.Text = ""  
txt_Quantity.Text = "0"  
txt_cOst.Text = "0.00"  
lbl_Amount.Caption = "0.00"  
lbl_Amount.Caption = "0.00"  
Txt_Remarks.Text = ""  
End Sub
```

```
Private Sub txt_Quantity_LostFocus()  
lbl_Amount = Val(txt_Quantity) * Val(txt_cOst)  
End Sub
```

```
Private Sub txt_Quantity_Validate(Cancel As Boolean)  
If IsNumeric(txt_Quantity) = False Then  
    MsgBox "Invalid quantity. Please enter a valid quantity. Numbers only.",  
    vbExclamation, "Invalid quantity"  
    txt_Quantity.SetFocus  
    SendKeys "{home}" & "+{end}"  
    Exit Sub  
End If  
If txt_Quantity < 1 Then  
    MsgBox "Please enter a valid quantity.", vbExclamation, "Invalid quantity"  
    Exit Sub  
End If  
End Sub
```

Change Password Form

```
Private Sub cmd_cancel_Click()  
Unload Me  
End Sub
```

```
Private Sub cmd_ok_Click()  
If UCase(Text1.Text) <> UCase(Text4.Text) Then  
    MsgBox "Invalid old-password.", , "Change Password"
```

```

Text1.SetFocus
SendKeys "{home}" & "+{end}"
Exit Sub
Else
If UCASE(Text2.Text) <> UCASE(Text3.Text) Then
    MsgBox "Please re-type the new password", , "Change Password"
    Text3.SetFocus
    SendKeys "{home}" & "+{end}"
    Exit Sub
Else
    Data1.Recordset.Edit
        Data1.Recordset.Fields(0)= Text2.Text
    Data1.Recordset.Update
    Data1.Refresh
    MsgBox "Your new password has been saved.", , "Password changed"
    Unload Me
End If
End If
End Sub

```

```

Private Sub Form_Load()
Data1.DatabaseName = App.Path & "\MICROFARM97.mdb"
Data1.RecordSource = "select * from Password"
Data1.Refresh
End Sub

```

Change User Name Form

```

Sub cmd_cancel_Click()
Unload Me
End Sub

```

```

Private Sub cmd_ok_Click()
If UCASE(Text5.Text) <> UCASE(Txt_dbChngUser.Text) Then
    MsgBox "Invalid old User Name.", , "Change User Name"
    Text5.SetFocus
    SendKeys "{home}" & "+{end}"
    Exit Sub
Else
If UCASE(Text6.Text) <> UCASE(Text7.Text) Then
    MsgBox "Please re-type the new User Name", , "Change User Name"
    Text7.SetFocus
    SendKeys "{home}" & "+{end}"
    Exit Sub
Else
    Data1.Recordset.Edit

```

```
Data1.Recordset.Fields(0) = Text6.Text  
Data1.Recordset.Update  
Data1.Refresh  
MsgBox "Your new User Name has been saved.", , "User Name changed"  
Unload Me  
End If  
End If  
End Sub
```

```
Private Sub Form_Load()  
Data1.DatabaseName = App.Path & "\MICROFARM97.mdb"  
Data1.RecordSource = "select * from Password"  
Data1.Refresh  
End Sub
```

Back-Up Database

```
Option Explicit  
Private Sub cmdSave_Click()  
    'saves a back-up copy of the system database  
    On Error Resume Next  
    Dim directory, Database  
    'get path  
    directory = dirlist1.Path  
    'if in root directory remove "\"  
    If Right(directory, 1) = Chr(92) Then directory = Left(directory, (Len(directory) - 1))  
    rtbdbase.FileName = App.Path + "\MICROFARM97.mdf"  
    'write the database file to disk  
    Database = FreeFile  
    Open directory + "\MICROFARM97_BackUp.mdb" For Output As Database  
    Print #1, rtbdbase.Text  
    Close Database  
    MsgBox "The back-up file was created."  
    'unload this form from memory  
    Unload Me  
End Sub
```

```
Private Sub cmdClose_Click()  
Unload Me  
End Sub
```

```
Private Sub drivelist_Change()  
    Me.dirlist1 = drivelist  
End Sub
```

```
Private Sub Form_Load()
```

```
Me.Top = 0  
Me.Left = 0  
End Sub
```

Egg Production Report

```
Option Explicit  
Dim StartStg As String  
Dim Endstg As String  
Dim l_results As ListItem  
Dim Total_Produced  
Dim rec_count, i  
Dim amt
```

```
Private Sub Populate_lvw_recordsfound()  
Dim rec_count, X  
  
With Data1.Recordset  
    .MoveLast  
    rec_count = .RecordCount  
    .MoveFirst  
    For X = 1 To rec_count  
        amt = Format(!Total_Produced, "#####")  
        Total_Produced = Total_Produced + Val(amt)  
        Set l_results = lvw_recordsfound.ListItems.Add(X, , !Date)  
        l_results.SubItems(1) = !Unit_1  
        l_results.SubItems(2) = !Unit_2  
        l_results.SubItems(3) = !Unit_3  
        l_results.SubItems(4) = !Total_Produced  
  
    .MoveNext  
    Next  
End With  
End Sub
```

```
Private Sub cmd_PreviewReport_Click()  
StartStg = DT_Start.Value  
Endstg = DT_End.Value  
If DataEnvironment1.rsCommand_EggProduction.State = adStateClosed Then  
    DataEnvironment1.rsCommand_EggProduction.Open  
    DataEnvironment1.rsCommand_EggProduction.Requery  
    DataEnvironment1.rsCommand_EggProduction.Requery  
    DataEnvironment1.Commands(1).CommandText = "select * from EggProduction  
WHERE ((EggProduction.Date) Between # " + StartStg + " # And # " + Endstg + " #)"  
    DataEnvironment1.Commands(1).Execute  
    DataEnvironment1.rsCommand_EggProduction.Requery
```

```
DataEnvironment1.rsCommand_EggProduction.Requery  
Load Rpt_EggProduction  
Rpt_EggProduction.Sections(1).Controls("Label5").Caption = "Date of report: " &  
StartStg & " to " & Endstg  
Rpt_EggProduction.Sections(5).Controls("label22").Caption  
Format(Total_Produced, "###,###,###.00")  
Rpt_EggProduction.Show vbModal  
If DataEnvironment1.rsCommand_EggProduction.State = adStateOpen Then  
DataEnvironment1.rsCommand_EggProduction.Close  
End Sub
```

```
Private Sub Command1_Click()
```

```
StartStg = DT_Start.Value  
Endstg = DT_End.Value  
lvw_recordsfound.ListItems.Clear  
Total_Produced = 0  
With Data1  
    .RecordSource = "select * from EggProduction WHERE (((EggProduction.Date)  
Between # " + StartStg + " # And # " + Endstg + " #))"  
    .Refresh  
    rec_count = .Recordset.RecordCount  
    If .Recordset.RecordCount <> 0 Then  
        Populate_lvw_recordsfound  
  
        cmd_PreviewReport.Enabled = True  
    Else  
        cmd_PreviewReport.Enabled = False  
    End If  
End With  
Label2.Caption = Format(Total_Produced, "###,###,###")  
End Sub
```

```
Private Sub Command2_Click()  
lvw_recordsfound.ListItems.Clear  
cmd_PreviewReport.Enabled = False  
Label2.Caption = "0"  
End Sub
```

```
Private Sub Form_Load()  
Data1.DatabaseName = App.Path & "\MICROFARM97.mdb"  
Data1.RecordSource = "select * from EggProduction"  
Data1.Refresh  
End Sub
```

```

Private Sub lvw_recordsfound_ColumnClick(ByVal ColumnHeader As
MSComctlLib.ColumnHeader)
    If lvw_recordsfound.SortOrder = lvwAscending Then
        lvw_recordsfound.SortOrder = lvwDescending
    Else
        lvw_recordsfound.SortOrder = lvwAscending
    End If
End Sub

```

Dropping Production Report

```

Option Explicit
Dim StartStg As String
Dim Endstg As String
Dim l_results As ListItem
Dim Total_Produced
Dim rec_count, i
Dim amt

```

```

Private Sub Populate_lvw_recordsfound()
Dim rec_count, X

With Data1.Recordset
    .MoveLast
    rec_count = .RecordCount
    .MoveFirst
    For X = 1 To rec_count
        amt = Format(!Number_of_bags, "#####")
        Total_Produced = Total_Produced + Val(amt)
        Set l_results = lvw_recordsfound.ListItems.Add(X, , !Date)
            l_results.SubItems(1) = !Unit_1
            l_results.SubItems(2) = !Unit_2
            l_results.SubItems(3) = !Unit_3
            l_results.SubItems(4) = !Number_of_bags
    .MoveNext
    Next
End With
End Sub

```

```

Private Sub cmd_PreviewReport_Click()
StartStg = DT_Start.Value
Endstg = DT_End.Value
If DataEnvironment1.rsCommand_DroppingsProduction.State = adStateClosed Then
DataEnvironment1.rsCommand_DroppingsProduction.Open
    DataEnvironment1.rsCommand_DroppingsProduction.Requery
    DataEnvironment1.rsCommand_DroppingsProduction.Requery

```

```

DataEnvironment1.Commands(1).CommandText      =      "select      *      from
DroppingsProduction WHERE (((DroppingsProduction.Date) Between # " + StartStg + "
# And # " + Endstg + "#)) "
DataEnvironment1.Commands(1).Execute
DataEnvironment1.rsCommand_DroppingsProduction.Requery
DataEnvironment1.rsCommand_DroppingsProduction.Requery
Load Rpt_EggProduction
Rpt_DroppingsProduction.Sections(1).Controls("Label5").Caption = "Date of report: "
& StartStg & " to " & Endstg
Rpt_DroppingsProduction.Sections(5).Controls("label22").Caption      =
Format(Total_Produced, "###,###,###.00")
Rpt_DroppingsProduction.Show vbModal
If DataEnvironment1.rsCommand_DroppingsProduction.State = adStateOpen Then
DataEnvironment1.rsCommand_DroppingsProduction.Close
End Sub

Private Sub Command1_Click()

StartStg = DT_Start.Value
Endstg = DT_End.Value
lvw_recordsfound.ListItems.Clear
Total_Produced = 0
With Data1
    .RecordSource = "select      *      from      DroppingsProduction      WHERE
((DroppingsProduction.Date) Between # " + StartStg + "# And # " + Endstg + "#)) "
    .Refresh
    rec_count = .Recordset.RecordCount
    If .Recordset.RecordCount <> 0 Then
        Populate_lvw_recordsfound

        cmd_PreviewReport.Enabled = True
    Else
        cmd_PreviewReport.Enabled = False
    End If
End With
Label2.Caption = Format(Total_Produced, "###,###,###")
End Sub

Private Sub Command2_Click()
lvw_recordsfound.ListItems.Clear
cmd_PreviewReport.Enabled = False
Label2.Caption = "0"
End Sub

Private Sub Form_Load()
Data1.DatabaseName = App.Path & "\MICROFARM97.mdb"

```

```

Data1.RecordSource = "select * from EggProduction"
Data1.Refresh
End Sub

Private Sub lvw_recordsfound_ColumnClick(ByVal ColumnHeader As MSComctlLib.ColumnHeader)
    If lvw_recordsfound.SortOrder = lvwAscending Then
        lvw_recordsfound.SortOrder = lvwDescending
    Else
        lvw_recordsfound.SortOrder = lvwAscending
    End If
End Sub

```

Sales Report Form

```

Option Explicit
Dim StartStg As String
Dim EndStg As String
Dim l_results As ListItem
Dim Total_sales
Dim rec_count, i
Dim amt

Private Sub Populate_lvw_recordsfound()
    Dim rec_count, X

    'Timer1.Enabled = True
    With Data1.Recordset
        .MoveLast
        rec_count = .RecordCount
        .MoveFirst
        For X = 1 To rec_count
            amt = Format(!Amount, "#####.00")
            Total_sales = Total_sales + Val(amt)
            Set l_results = lvw_recordsfound.ListItems.Add(X, , !OR_Number)
            l_results.SubItems(1) = !Date_Sold
            l_results.SubItems(2) = !Customer_Name
            l_results.SubItems(3) = !Item_Description
            l_results.SubItems(4) = !Unit_Price
            l_results.SubItems(5) = !Qty
            l_results.SubItems(6) = !Amount

        .MoveNext
        Next
    End With
End Sub

```

```

Private Sub cmd_PreviewReport_Click()
StartStg = DT_Start.Value
Endstg = DT_End.Value
If DataEnvironment1.rsCommand_Sales.State = adStateClosed Then
DataEnvironment1.rsCommand_Sales.Open
DataEnvironment1.rsCommand_Sales.Requery
DataEnvironment1.rsCommand_Sales.Requery
DataEnvironment1.Commands(1).CommandText = "select * from Sales WHERE
((Sales.Date_Sold) Between # " + StartStg + " # And # " + Endstg + " #))"
DataEnvironment1.Commands(1).Execute
DataEnvironment1.rsCommand_Sales.Requery
DataEnvironment1.rsCommand_Sales.Requery
Load Rpt_Sales
Rpt_Sales.Sections(1).Controls("Label5").Caption = "Date of report: " & StartStg &
to " & Endstg
Rpt_Sales.Sections(5).Controls("label22").Caption = Format(Total_sales,
"###,###,##.00")
Rpt_Sales.Show vbModal
If DataEnvironment1.rsCommand_Sales.State = adStateOpen Then
DataEnvironment1.rsCommand_Sales.Close
End Sub

Private Sub Command1_Click()
StartStg = DT_Start.Value
Endstg = DT_End.Value
lvw_recordsfound.ListItems.Clear
Total_sales = 0
With Data1
    .RecordSource = "select * from Sales WHERE ((Sales.Date_Sold) Between # " +
StartStg + " # And # " + Endstg + " #))"
    .Refresh
    rec_count = .Recordset.RecordCount
    If .Recordset.RecordCount <> 0 Then
        Populate_lvw_recordsfound
        cmd_PreviewReport.Enabled = True
    Else
        cmd_PreviewReport.Enabled = False
    End If
End With
Label2.Caption = Format(Total_sales, "###,###,##.00")
End Sub

Private Sub Command2_Click()
lvw_recordsfound.ListItems.Clear

```

```
cmd_PreviewReport.Enabled = False  
Label2.Caption = "0.00"  
End Sub
```

```
Private Sub Form_Load()  
Data1.DatabaseName = App.Path & "\MICROFARM97.mdb"  
Data1.RecordSource = "select * from Sales"  
Data1.Refresh  
End Sub
```

```
Private Sub lvw_recordsfound_ColumnClick(ByVal ColumnHeader As MSComctlLib.ColumnHeader)  
If lvw_recordsfound.SortOrder = lvwAscending Then  
    lvw_recordsfound.SortOrder = lvwDescending  
Else  
    lvw_recordsfound.SortOrder = lvwAscending  
End If  
End Sub
```

Expenditure Report Form

```
Option Explicit  
Dim StartStg As String  
Dim Endstg As String  
Dim l_results As ListItem  
Dim Total_Purchases  
Dim rec_count, i  
Dim amt
```

```
Private Sub cmd_PreviewReport_Click()  
StartStg = DT_Start.Value  
Endstg = DT_End.Value  
If DataEnvironment1.rsCommand_Purchases.State = adStateClosed Then  
    DataEnvironment1.rsCommand_Purchases.Open  
    DataEnvironment1.rsCommand_Purchases.Requery  
    DataEnvironment1.rsCommand_Purchases.Requery  
    DataEnvironment1.Commands(2).CommandText = "select * from Expenditure  
WHERE ((Expenditure.Date_purchased) Between # " + StartStg + " # And # " + Endstg  
+ " #))"  
    DataEnvironment1.Commands(2).Execute  
    DataEnvironment1.rsCommand_Purchases.Requery  
    DataEnvironment1.rsCommand_Purchases.Requery  
    Load Rpt_Purchases  
    Rpt_Purchases.Sections(1).Controls("Label5").Caption = "Date of report: " & StartStg  
& " to " & Endstg
```

```
Rpt_Purchases.Sections(5).Controls("label22").Caption = Format(Total_Purchases,  
"###,###,##.00")  
Rpt_Purchases.Show vbModal  
If DataEnvironment1.rsCommand_Purchases.State = adStateOpen Then  
DataEnvironment1.rsCommand_Purchases.Close  
End Sub
```

```
Private Sub cmd_Clear_Click()  
lvw_recordsfound.ListItems.Clear  
cmd_PreviewReport.Enabled = False  
End Sub
```

```
Private Sub cmd_Search_Click()  
StartStg = DT_Start.Value  
EndStg = DT_End.Value  
lvw_recordsfound.ListItems.Clear  
Total_Purchases = 0  
With Data1  
.RecordSource = "select * from expenditure WHERE (((Expenditure.Date_purchased)  
Between # " + StartStg + " # And # " + EndStg + " #))"  
.Refresh  
rec_count = .Recordset.RecordCount  
If .Recordset.RecordCount <> 0 Then  
    Populate_lvw_recordsfound  
  
    cmd_PreviewReport.Enabled = True  
Else  
    cmd_PreviewReport.Enabled = False  
End If  
End With  
Label2.Caption = Format(Total_Purchases, "###,###,##.00")  
End Sub
```

```
Private Sub Form_Load()  
Data1.DatabaseName = App.Path & "\MICROFARM97.mdb"  
Data1.RecordSource = "select * from Expenditure"  
Data1.Refresh  
End Sub
```

```
Private Sub Populate_lvw_recordsfound()  
Dim rec_count, X
```

```
With Data1 .Recordset  
.MoveLast  
rec_count = .RecordCount  
.MoveFirst
```

```

For X = 1 To rec_count
amt = Format(!Amount, "#####00")
Total_Purchases = Total_Purchases + Val(amt)
Set l_results = lvw_recordsfound.ListItems.Add(X, !purchase_code)
    l_results.SubItems(1) = !Date_purchased
    l_results.SubItems(2) = !Type
    l_results.SubItems(3) = !Item_Description
    l_results.SubItems(4) = !Qty
    l_results.SubItems(5) = !cost
    l_results.SubItems(6) = !Amount
.MoveNext
Next
End With
End Sub

```

```

Private Sub lvw_recordsfound_ColumnClick(ByVal ColumnHeader As MSComctlLib.ColumnHeader)
If lvw_recordsfound.SortOrder = lvwAscending Then
    lvw_recordsfound.SortOrder = lvwDescending
Else
    lvw_recordsfound.SortOrder = lvwAscending
End If
End Sub

```

Help Form

Option Explicit

```

'Reg Key Security Options...
Const READ_CONTROL = &H20000
Const KEY_QUERY_VALUE = &H1
Const KEY_SET_VALUE = &H2
Const KEY_CREATE_SUB_KEY = &H4
Const KEY_ENUMERATE_SUB_KEYS = &H8
Const KEY_NOTIFY = &H10
Const KEY_CREATE_LINK = &H20
Const KEY_ALL_ACCESS = KEY_QUERY_VALUE + KEY_SET_VALUE +
    KEY_CREATE_SUB_KEY + KEY_ENUMERATE_SUB_KEYS +
    KEY_NOTIFY + KEY_CREATE_LINK + READ_CONTROL

```

```

'Reg Key ROOT Types...
Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1          ' Unicode nul terminated string
Const REG_DWORD = 4       ' 32-bit number

```

```
Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location"
Const gREGVALSYSINFOLOC = "MSINFO"
Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"
Const gREGVALSYSINFO = "PATH"

Private Declare Function RegOpenKeyEx Lib "advapi32" Alias "RegOpenKeyExA"
(ByVal hKey As Long, ByVal lpSubKey As String, ByVal ulOptions As Long, ByVal
samDesired As Long, ByRef phkResult As Long) As Long
Private Declare Function RegQueryValueEx Lib "advapi32" Alias "RegQueryValueExA"
(ByVal hKey As Long, ByVal lpValueName As String, ByVal lpReserved As Long,
ByRef lpType As Long, ByVal lpData As String, ByRef lpcbData As Long) As Long
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long) As Long
```

```
Private Sub cmdSysInfo_Click()
    Call StartSysInfo
End Sub
```

```
Private Sub cmdOK_Click()
    Unload Me
End Sub
```

```
Private Sub Form_Load()
    Me.Caption = "About " & App.Title
    lblVersion.Caption = "Version " & App.Major & "." & App.Minor & "." &
App.Revision
    lblTitle.Caption = "MicroFarm"
End Sub
```

```
Public Sub StartSysInfo()
    On Error GoTo SysInfoErr

    Dim rc As Long
    Dim SysInfoPath As String

    ' Try To Get System Info Program Path\Name From Registry...
    If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO,
gREGVALSYSINFO, SysInfoPath) Then
        ' Try To Get System Info Program Path Only From Registry...
        ElseIf GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFOLOC,
gREGVALSYSINFOLOC, SysInfoPath) Then
            ' Validate Existance Of Known 32 Bit File Version
            If (Dir(SysInfoPath & "\MSINFO32.EXE") <> "") Then
                SysInfoPath = SysInfoPath & "\MSINFO32.EXE"

            ' Error - File Can Not Be Found...
        End If
    End If
End Sub
```

```

    Else
        GoTo SysInfoErr
    End If
' Error - Registry Entry Can Not Be Found...
Else
    GoTo SysInfoErr
End If

Call Shell(SysInfoPath, vbNormalFocus)

Exit Sub
SysInfoErr:
    MsgBox "System Information Is Unavailable At This Time", vbOKOnly
End Sub

Public Function GetKeyValue(KeyRoot As Long, KeyName As String, SubKeyRef As
String, ByRef KeyVal As String) As Boolean
    Dim i As Long                      ' Loop Counter
    Dim rc As Long                      ' Return Code
    Dim hKey As Long                    ' Handle To An Open Registry Key
    Dim hDepth As Long
    Dim KeyValType As Long              ' Data Type Of A Registry Key
    Dim tmpVal As String                ' Temporary Storage For A Registry Key
Value
    Dim KeyValSize As Long              ' Size Of Registry Key Variable
'-----
' Open RegKey Under KeyRoot {HKEY_LOCAL_MACHINE...}
'-----
    rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey) ' Open
Registry Key

    If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError      ' Handle Error...

    tmpVal = String$(1024, 0)                      ' Allocate Variable Space
    KeyValSize = 1024                                ' Mark Variable Size

'-----
' Retrieve Registry Key Value...
'-----
    rc = RegQueryValueEx(hKey, SubKeyRef, 0, _
        KeyValType, tmpVal, KeyValSize)   ' Get/Create Key Value

    If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError      ' Handle Errors

    If (Asc(Mid(tmpVal, KeyValSize, 1)) = 0) Then      ' Win95 Adds Null Terminated
String...

```

```
tmpVal = Left(tmpVal, KeyValSize - 1)      ' Null Found, Extract From String
Else                                         ' WinNT Does NOT Null Terminate String...
    tmpVal = Left(tmpVal, KeyValSize)        ' Null Not Found, Extract String
End If
```

Determine Key Value Type For Conversion...

```
Select Case KeyValType                  ' Search Data Types...
Case REG_SZ                            ' String Registry Key Data Type
    KeyVal = tmpVal                   ' Copy String Value
Case REG_DWORD                          ' Double Word Registry Key Data Type
    For i = Len(tmpVal) To 1 Step -1   ' Convert Each Bit
        KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1))) ' Build Value Char. By Char.
    Next
    KeyVal = Format$("&h" + KeyVal)     ' Convert Double Word To String
End Select
```

```
KeyValue = True                         ' Return Success
rc = RegCloseKey(hKey)                  ' Close Registry Key
Exit Function                           ' Exit
```

```
KeyError:    ' Cleanup After An Error Has Occured...
    KeyVal = ""                      ' Set Return Val To Empty String
    KeyValue = False                 ' Return Failure
    rc = RegCloseKey(hKey)          ' Close Registry Key
End Function
```