

**CHRISTIAN SERVICE UNIVERSITY COLLEGE,
KUMASI**



**DEPARTMENT OF COPUTER SCIENCE
B.S.C COMPUTER SCIENCE**

**COLD STORE MANAGEMENT SYSTEM
CASE STUDY: CHRISTIAN SERVICE UNIVERSITY (COMPUTER SCIENCE
DEPARTMENT)**

**BY
BANNERMAN CHARLES JAMES**

**A LONG ESSAY PRESENTED TO CHRISTIAN SERVICE UNIVERSITY COLLEGE IN
PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF
B.S.C COMPUTER SCIENCE**

JUNE, 2012

DECLARATION

I, Bannerman Charles James hereby declare that this piece of work was undertaken by me. I further affirm that this work has never been submitted to any University in part or whole for a diploma, degree or any award. That the work of which it is a record has been done by me. All quotations have been distinguished either by quotation marks or indentations in the body of the essay. The source of information has specifically been acknowledged by means of footnotes.

Student's Name

Signature

Date

Bannerman Charles James

.....

.....

Supervisor's Name

Miss. Linda Amoako

.....

.....

Head of Department's Name

Miss Linda Amoako

.....

.....

DEDICATION

I dedicate this work to the sustainer of life, God Almighty, my family and all loved ones.

ACKNOWLEDGEMENTS

I give glory and honour to the Lord Almighty who has shown to me his unfailing love. Am also very thankful to Mr. Linda Amoako who took patience and pain to read and correct the script severally with the provision of necessary information and suggestions for the success of this study.

I am grateful to Mr. Bruce Akwesi Gyane and Sheila Gyamfi who assisted me in the interviews for the study and gave me all the support I needed to make this study a success. I appreciate Mr. Joseph Owusu Ansah and Mrs. Rosina Amankwah who helped by providing the necessary information needed for the study.

I sincerely express my gratitude to my family and all friends and loved ones who gave me the encouragement towards the completion of this research work.

pages

Dedication.....i

Acknowledgementsii

Table of contents..... v-
vii

CHAPTER ONE: INTRODUCTION

1.1 Introduction.....1

1.2 Problem identification.....2

1.2 Aim of the project.....3

1.4 Research objectives.....3

1.5 Methodology.....3

1.6 Relevance of the study4

1.7 Scope of the project.....4

CHAPTER TWO: LITERATURE REVIEW

2.0 Introduction.....5

2.1 information system.....6

2.2 Steps in developing management information system.....6-8

CHAPTER THREE: REQUIREMENT AND DESIGN SPECIFICATION

3.1 Introduction.....9

3.1 Data collection.....9

3.2 Functional requirement.....9

3.3 Feasibility study.....10

3.4 Economic feasibility.....10

3.5 Functional specification of the proposed system.....11

3.6 Design specification.....11

3.7 Database design.....12-14

3.8 User interface design requirement.....15-16

CHAPTER FOUR: IMPLEMENTATION OF SOLUTION

4.0 Testing.....17

4.1 Unit testing.....	17
4.2 Integration testing.....	17-28
4.3 Evaluating the project.....	29
4.4 Evaluating the solution.....	29
4.5 Acceptance testing.....	29
4.6 Data take-on and conversion.....	29

CHAPTER FIVE: CONCLUSION

5.0 Summary of problems, achievements and challenges.....	30
5.1 Problems.....	30
5.2 Achievements.....	30
5.3 Recommendation.....	30
References.....	31
Appendix.....	32-40

CHAPTER ONE

1.0 INTRODUCTION

Software development in the world, a much used and more generic term, does not necessarily subsume the engineering paradigm. Although it is questionable what impact it has had on actual software development over the last 40 years, the field's future looks bright according to (Money Magazine,2006; Salary.com,) which rated "Software Engineer" as the best job in the United States in 2006(Wikipedia.org).

In Ghana, the development of modern Information Systems is a complex interdisciplinary process. The developer and the user must be knowledgeable in their respective areas of functions and responsibilities. Furthermore, different technologies, other than information technology, are used in organizations for providing input to the information systems. As the organization processes are getting automated, the information systems are undergoing changes which are good towards making them more suited to the organizations' needs.

Cold store management includes functionalities from many types of software, including monitoring and management. It is on this view that the Cold Store Management System is being designed to solve the problems faced in relation to keeping records at Chilling Cold Store in Kumasi Asafo market.

The world is gradually moving from the age whereby processing of data was done using manual methods and filing systems to the age of processing information with the help of new technologies through the use of computers and devices like mobile phones. The record keeping routines, information tracking, complicated logistics and many other administrative procedures are all regular activities that constantly make up the core processes in the management of real and financial assets. However these processes could become quite cumbersome and inefficient when manual systems are being used as they play prominent roles in organizations.

With the advent of Information Technology (IT), some organizations and individuals have employed various database management systems for the management of assets. These systems

are generic hence there is need to design a system that is tailored to specific needs of asset management

This Project basically seeks to implement the right change by making less use of the manual system and to automate the company's data that are kept in files in a computerized database so as to make information retrieval and enhance data security. It has been grouped into two modules to ensure that all information needed by the Company are available in the database. The first module takes care of the Cold Store management system by keeping information on all transactions made since the establishment of the company, show transaction status, the employees that logged into the system and the date under which the transactions carried out. The second module system has a security facility which enables only authorized users to gain access to the system, thereby offering protection of information.

1.1 PROBLEM IDENTIFICATION:

The Company is currently using the manual method of managing its transaction and is faced with numerous challenges.

-) The files in which the data are stored in are not kept well. Its confidentiality is compromised due to the unsecured storage of the data and retrieval of information becomes a problem.
-) The company uses the manual way of keeping their records. Since these papers are important and will be necessary for future use. They have to be retained for a while and this makes office full of papers.
-) The tendency of making mistakes is very high when dealing with paper work. Errors can be made by employees and can make auditing of the company's records very difficult to handle and to interpret.

1.2 AIM OF THE PROEECT

The main aim of the project is to move from the manual ways of keeping records and making the records more secured in a database.

1.3 RESEARCH OBJECTIVES

The objectives of the project are as follows;

-) To improve upon the current existing manual system and replace it with a more efficient software.
-) To be able to obtain detailed information on all work and keep them in one database to make information retrieval easy.
-) To be able to monitor the status of work under particular year.

1.4 METHODOLOGY

In rapid application development the primary importance is on building a prototype that looks and acts like the desired product in order to test its usefulness. The prototype is a vital part of the requirement determination phase, and may be created using the tools different from those used for the final product.

Once the prototype is approved, it is discarded and the real software is written. These steps may include:

-) Requirement gathering.
-) Creating design.
-) Building prototype.
-) Customer evaluation.
-) Refining prototype and repeat the steps.
-) Engineer product.

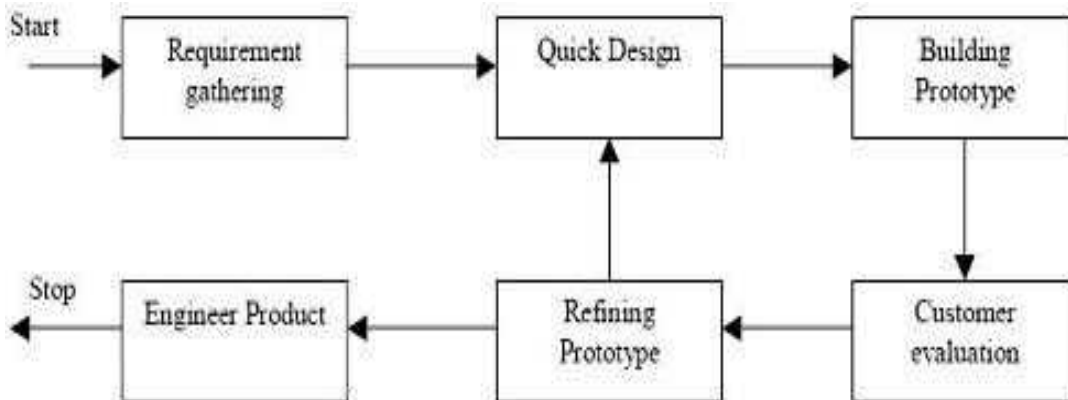


Figure 1.0 Source: Software Engineering by Sabharwal Sangeeta

1.5 RELEVANCE OF THE STUDY

Implementation of software has solved most problems in organizations. The importance of this software is to help the company solve the principal problems with bookkeeping and asset management. Since all processes are going to be computerized, it going to save time and energy, reduce the paper work and make auditing easier.

1.6 SCOPE OF THE PROJECT

This system will focus on sea frozen food Companies with specific reference to Chilling Cold Store which can be located at Asafo market in Kumasi.

With respect to the case study, the company is facing a problem with managing their records with regards to the use of manual system. The company has a lot of files packed in the office.

The proposed system will authenticate the users of the system and will monitor who logs in at a point in time and out of the system. The proposed system will enable easy storage, searching and retrieval of information. The system will back up all the transactions which have been in the system for long time. Help to obtain detailed information on records and keep them in one database to make information retrieval easy.

CHAPTER TWO

LITERATURE REVIEW

2.0 INTRODUCTION

The purpose of this literature review is to summarize available research that is related to the concept of Cold Store Management Systems and the evaluation of the efficiency and performance of these systems in the Company. A second section will take snap shots into record management.

Asset management is an emerging effort to integrate finance, planning, engineering, personnel, and information management to assist agencies in managing assets cost-effectively. In its broadest sense, asset management is defined as “a systematic process of maintaining, upgrading, and operating assets, combining engineering principles with sound business practice and economic rationale, and providing tools to facilitate a more organized and flexible approach to making the decisions necessary to achieve the public’s expectations” Organization for Economic Cooperation Development (OECD 2001). The main objective of asset management is to improve decision-making processes for allocating funds among an agency’s assets so that the best return on investment is obtained. To achieve this objective, asset management embraces all of the processes, tools, and data required to manage assets effectively (Nemmers, 2004). For this reason asset management is also defined as “a process of resource allocation and utilization. The framework needed to carry out this process effectively encompasses an agency’s policy goals and objectives, performance measurements, planning and programming, program delivery, and system monitoring and performance results.

Corporate accountants also known as management accountants, perform a series of tasks to ensure their company's financial security, handle all financial matters and help to drive the business overall management and strategy. A management accountant's responsibilities can range widely. Depending on the company, one’s level of experience, the time of year and the type of industry, one could find oneself doing anything from budgeting, handling taxes and managing assets to helping determine compensation and benefits packages and aiding in strategic planning.

Records Management is the practice of identifying, classifying, archiving, preserving, and sometimes destroying records. According to International Standard Organization (ISO, 2001). (ISO, 2001) records are defined as “information created, received, and maintained as evidence and information by an organization or person, in pursuance of legal obligations or in the transaction of business”. The International Council on Archives (ICA, 2009) and Electronic Records define a record as,” a specific piece of recorded information generated, collected or received in the initiation, conduct or completion of an activity and which comprises sufficient content, context and structure to provide proof or evidence of that activity”.

While the definition of a record is often identified strongly with a document, a record can be either a tangible object or digital information which has value to an organization. For example, office documents, databases and application data, and e-mail are all examples of records.

2.1 INFORMATION SYSTEM

O’Brien, (2002) Defines Information System as any organized combination of people, hardware, software, communication networks and data resources, that control, transform and disseminate information in an organization. O’Brien, (2002) further reveals that the data resources of information systems are typically organized, stored and accessed by a variety of data resource management technologies into:

1. Databases that hold processed and organized data.
2. Knowledge bases that hold knowledge in a variety of forms such as facts, rules, and case examples about successful business practices.

2.2 STEPS IN DEVELOPING MANAGEMENT INFORMATION SYSTEM

These are the following steps for developing management information system:

- (a) Review the existing system.
- (b) Define the data needs for relevant units within the system.
- (c) Determine the most appropriate and effective data flow.

- (d) Design the data collection and reporting tools.
- (e) Develop the procedures and mechanisms for data processing.
- (f) Develop and implement a training program for data providers and data users.
- (g) Pre-test, and if necessary re-design the system for data collection, data flow, data processing and data utilization.
- (h) Monitor and evaluate the system.
- (i) Develop effective data dissemination and feedback mechanisms.
- (j) Evaluate the system.

According to Chrisanthi and Cornford, (1996), the process of development of an Information System can be seen as a list of tasks, starting with identification and launching of an information system's project and ending with maintenance of its optional components for a period before the system is phased out or replaced. However, he says that this varies from one organization description to another. On their part they suggest the following steps to be typical for most organizations:

- (a) Identification of a problem, pressure, or opportunity
- (b) Determination of general requirements for change
- (c) Feasibility study to explore possible approaches
- (d) Systems analysis to model detailed requirements for technical components or organizational reform
- (e) Systems design to work out how requirements are to be met
- (f) Development or acquisition of software and hardware and their configuration
- (g) Systems implementation with the organizational settings
- (h) Operation and maintenance
- (i) Phase out when the system is no longer needed or used.

Although many scholars have written about computerization of records as an urgent requirement and a lot of research has been carried out, implementation is still insufficient. However, according to the literature available, there are numerous benefits that accrue from the proposed

system when compared with manual systems. For example there will be no duplication of records, the problem of missing and / or misplaced records are reduced and the information is available at the point of care.

CHAPTER THREE

REQUIREMENT AND DESIGN SPECIFICATION

3.0 INTRODUCTION

Requirement Analysis is the process of gathering and interpreting facts, weighing problems critically and using the information to propose improvements to the system.

The System Analysis is conducted with the following basic objectives in mind:

1. Requirement of the client's needs.
2. Evaluate the system concept for feasibility.
3. Perform economical and technical analysis.
4. Allocate functions of hardware, software, people, database and other system elements.
5. Adjust cost and plan constraints.

3.1 DATA COLLECTION

In gathering the needed requirement for this project, a number of visits were made to Chilling Company Limited. The purpose for the visits was aimed at finding out the requirements needed for the developing of the cold store management system. Interviewing was the techniques used in the requirement gathering. Series of interviews were made with some the workers in the company. This method was adapted because it gives accurate and carefully documented data and also minimise cost and less time-consuming.

3.2 FUNCTIONAL REQUIREMENTS

These are necessary conditions that must prevail in order for the proposed system to function effectively because they state what the company wants in the system (i.e. services). In summary the functional requirements are:

1. Keeping track of all daily transactions.
2. Being able to give accurate report.
3. No unauthorized person can use the system.

4. Changes in database can only be done by the administrators.

3.3 FEASIBILITY STUDY

Feasibility study is a preliminary study which investigates the information needs of prospective users and determines the resource requirements, costs and benefits of a proposed project. Feasibility is also concerned with determining the cost effectiveness of various alternatives in the designs of the information system and the priorities among the various system components.

3.4 ECONOMIC FEASIBILITY

Economic feasibility is the measure of the cost effectiveness of a project. It is also known as cost-benefit analysis. Economically, the system should be a good investment to the organization; the benefits must either be equal or exceed the cost of development.

Economic feasibility covers the following areas:

-) the cost to conduct a full systems investigation
-) the cost of hardware and software for the class of application being considered
-) the benefits in the form of reduced costs or fewer costly errors
-) the cost if nothing changes (i.e. the proposed system is not developed)

In the development of the system, the feasibility study was conducted within that company alone, all reports and documents were provided by the appropriate personnel. Due to this the cost of conducting the study was negligible. Hardware and software with the expected specifications were also readily available. As a result the system passes the test of Economic feasibility.

3.5 FUNCTIONAL SPECIFICATION OF THE PROPOSED SYSTEM

HARDWARE REQUIREMENTS

The host computer system should have the basic components of any computer system (keyboard, mouse, system unit and monitor) and should be IBM or IBM Compatible. Additionally, it should have:

1. A minimum of 512MB of RAM recommended.
2. Intel® processor Pentium IV 900MHz or equivalent or higher.
3. A minimum of 1.0GB free space on hard disk recommended.

SOFTWARE REQUIREMENTS

The host computer should be running preferably on Microsoft Windows® XP Professional Edition or Home Edition or Microsoft® Windows 2000 operating system with service pack 2. These OS are very stable and quite strong with respect to security. It can serve as good platforms for server operations as well as for networking.

Additionally, it should have the Microsoft.Net Framework 1.0. This is the framework on which all programmers written with any of the .Net family of languages can run.

The host computer should have SQL Server 2005 or later installed as this is the database server to be used.

3.6 DESIGN SPECIFICATION

The design specification gives the precise and accurate information about the requirement for the system design, which will help the user to gain the intended outcome.

3.7 DATABASE DESIGN

Almost all modern businesses involve storing and processing information. To address this need a logical data model should be developed. This involves outlining distinct entities such as customer details, debtor details and project data and formally defining relationships between them. Once a logical data model is designed, database management tools such as Microsoft SQL Server, Oracle, Microsoft Access, Sybase Server or Informix Server can be used to implement that model in the form of a database and then develop the appropriate client application using products such as Microsoft Visual Studio to support it.

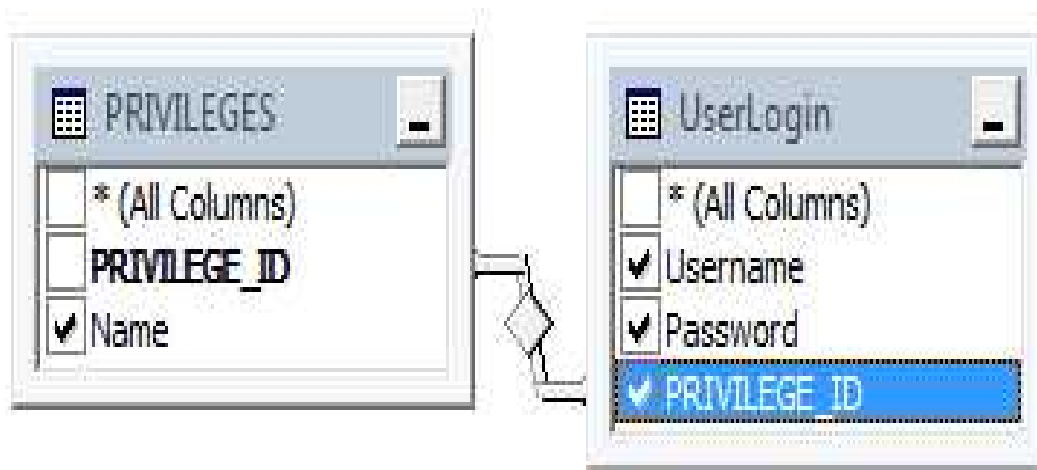


TABLE 3.1 USER LOGIN RELATION

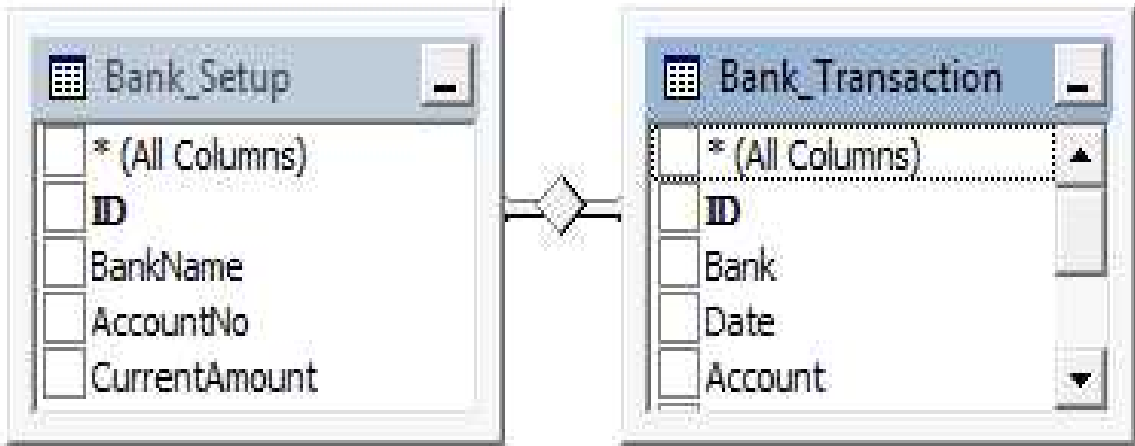


TABLE 3.2 BANK SETUP RELATION

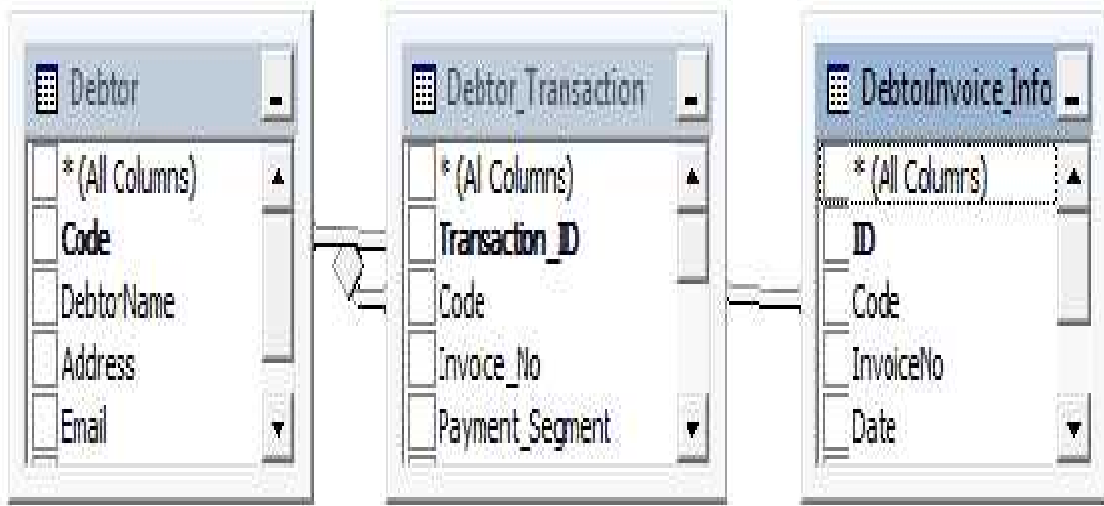


TABLE 3.3 DEBTOR RELATION

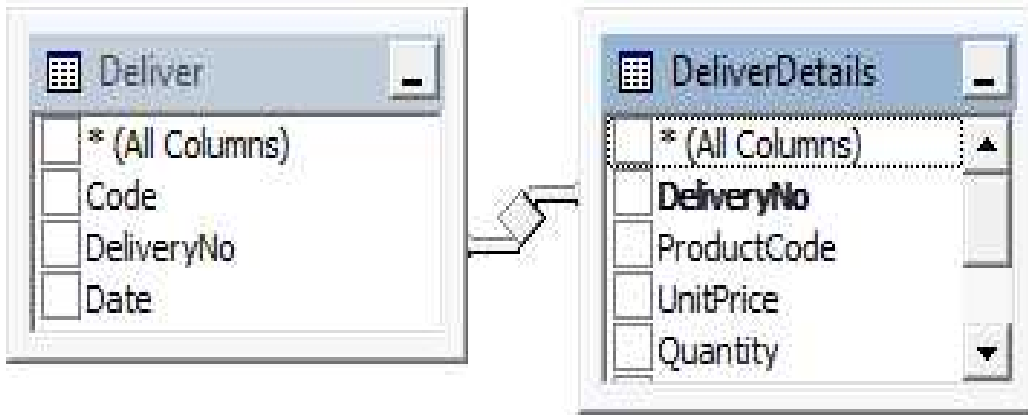


TABLE 3.4 DELIVERY RELATION

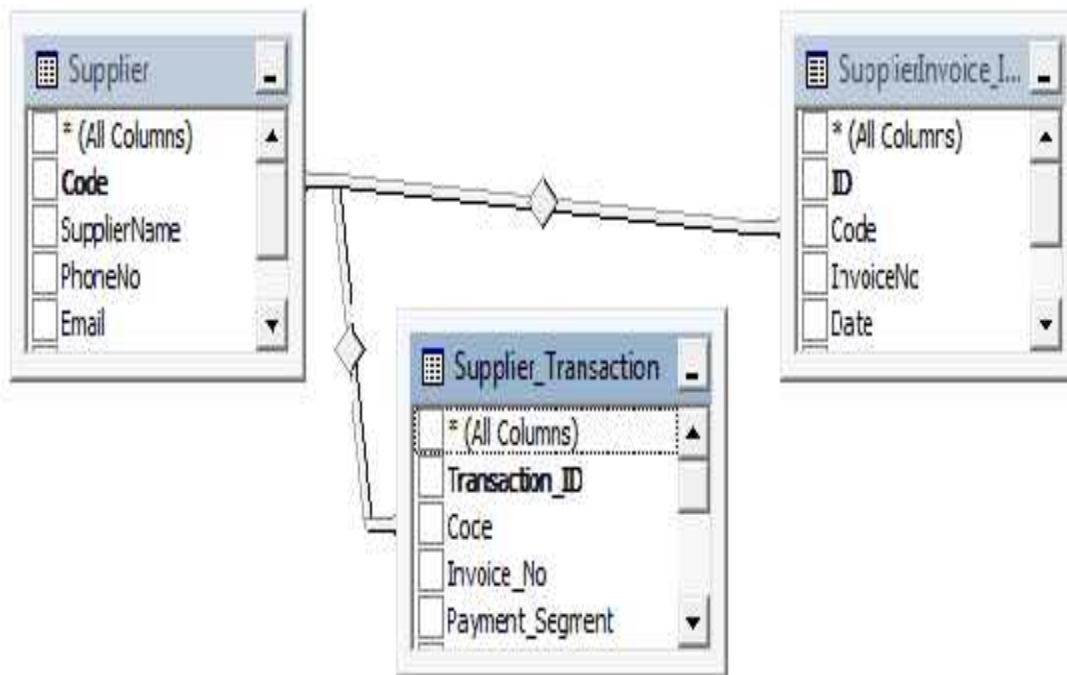


TABLE 3.5 SUPPLIER RELATION

3.8 USER INTERFACE DESIGN REQUIREMENT

Input design consists of developing specifications and procedures for data preparation, those steps necessary to put transaction data into a useable form for processing, data entry, the activity of outputting that data into the computer for processing. Some main objectives guiding the design of input includes controlling the amount of input required, avoiding delay, controlling errors, and keeping the steps simple. The graphical user interface (GUI) forms below are to be used by both the users and the administrator



Figure 3.6 Login Form

The figure 3.6 is produced when the program is being run to execute. Users can login with a provided username and password. This will pop up a welcome form as 3.6.

1. The OK button enables the users to log into the system by providing a valid user name and a secured password.
2. The change password button gives an option to the administrator to change the password of users of the system.

3. The cancel button closes the form.



Figure 3.7 Welcome Form.

The figure 3.7 helps link all components in the database to their corresponding table in the database. When this step is completed successfully. The main form comes as shown in the figure 3.7.

CHAPTER FOUR

IMPLEMENTATION OF SOLUTION

4.0 TESTING

The process of checking whether the developed system is working according to the original requirements and objectives is known as system testing. There are basically two sources of data for testing. These are Live and Artificial. Live data are actually extracted from the organization's files. Artificial data are generated for test purposes only.

The purpose of system testing is to test the system exhaustively including any additional housekeeping functions like file archiving. It is the application developer's last opportunity to check that the system works before asking the client to accept it. Therefore, it is often split between test running and bug fixing. This helps ensure adequate control over the way corrections are applied to the system and system test version control documents are kept. System testing incorporates a number of other classes of testing.

4.1 UNIT TESTING

The first level of test is unit testing. The purpose of unit testing is to ensure that each subprogram (subroutine) is fully tested. To successfully test a system the programmer writes a test plan. A test Plan is a document that describes the objectives, scope, approach, and focus of a software testing effort. This includes a number of test runs such as the valid paths through the code, the exception and error handling paths.

4.2 INTEGRATION TESTING

The next step is integration testing. This is sometime called link, subsystem or level one testing. The purpose of integration testing is to test the interface between programs in the same functional area. The output from unit testing becomes the input to integration testing. Each program is linked to the other entire program with which it interacts.

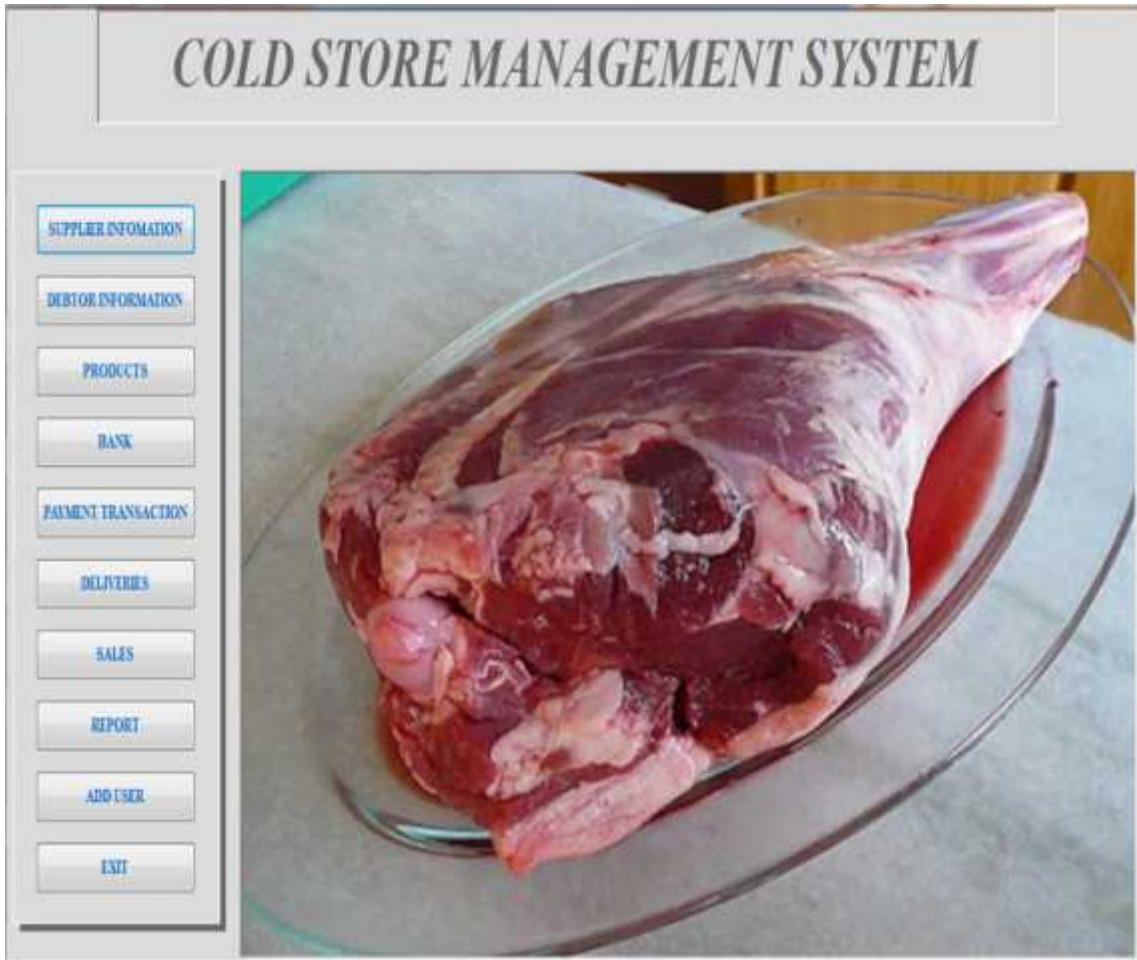


Figure 4.1 Main form.

When button 1 suppliers informaton is clicked, the suppliers database Form appears. When button 2 debtor information is clicked, the debtor database Form appears. When button 3 bank is clicked, the bank transaction Form appears. When button 4 payment transaction is clicked, the payment transaction Form appears. When button 5 deliveries is clicked, the deliveries form appears. When button 6 sales transaction is clicked the sales form appear. The same thing applies to rest of the buttons they clicked. The named forms are shown below;

The image shows a web-based form titled "Supplier Database". The form is divided into several sections. On the right side, there is a "Search" section with a dropdown menu and a "Refresh" button. The main form area contains several input fields: "Supplier Code", "Email Address", "Company Name", "Fax Number", "Phone Number", and "Address". At the bottom of the form, there are five buttons: "Add", "Modify", "Save", "Delete", and "Cancel". A small image of fish is visible in the bottom right corner of the form area.

Figure 4.2 Suppliers Form

The supplier's database consists of the following information about a supplier which will be stored in the database. Each supplier is given a unique supplier code which is followed by the company's name and other necessary information about a supplier such as the phone number, fax number, address and an e-mail address. In this figure 4.2, user can make a search for its existing suppliers as in figure 4.2 and can make any modification to an existing supplier's information.

-) The **Add** button is clicked to add information about new supplier.
-) The **Modify** button is clicked to make any changes on information about the supplier.
-) The **Save** button saves information about the supplier into the database.
-) The **Delete** button deletes the information about the supplier from the database.
-) The **Cancel** button closes the form.

The image shows a web-based form titled "Debtor Database". The form is divided into several sections. On the left side, there are five input fields for data entry: "Debtor Code", "Name", "E-mail Address", "Phone Number", and "Address". On the right side, there is a search section with a dropdown menu and a "Refresh" button. Below the search section, there is a row of five buttons: "Add", "Modify", "Save", "Delete", and "Cancel". In the top right corner, there is a small image of a hand holding a pen over a document with the URL "www.alibaba.com" visible.

Figure 4.3 Debtor information form.

The debtor's database consists of the following information about a debtor which will be stored in the database. Each supplier is given a unique debtor code which is followed by the debtor's name and other necessary information about a debtor such as the phone number, fax number, address and an e-mail address. On this figure 4.3, user can make a search for its existing debtors as shown above and can make any modification to an existing s debtor's information.

-) The **Add** button is clicked to add information about new debtor.
-) The **Modify** button is clicked to make any changes on information about the debtor.
-) The **Save** button saves every information about the debtor.
-) The **Delete** button deletes the information about the debtor from the database.
-) The **Exit** button closes the form.

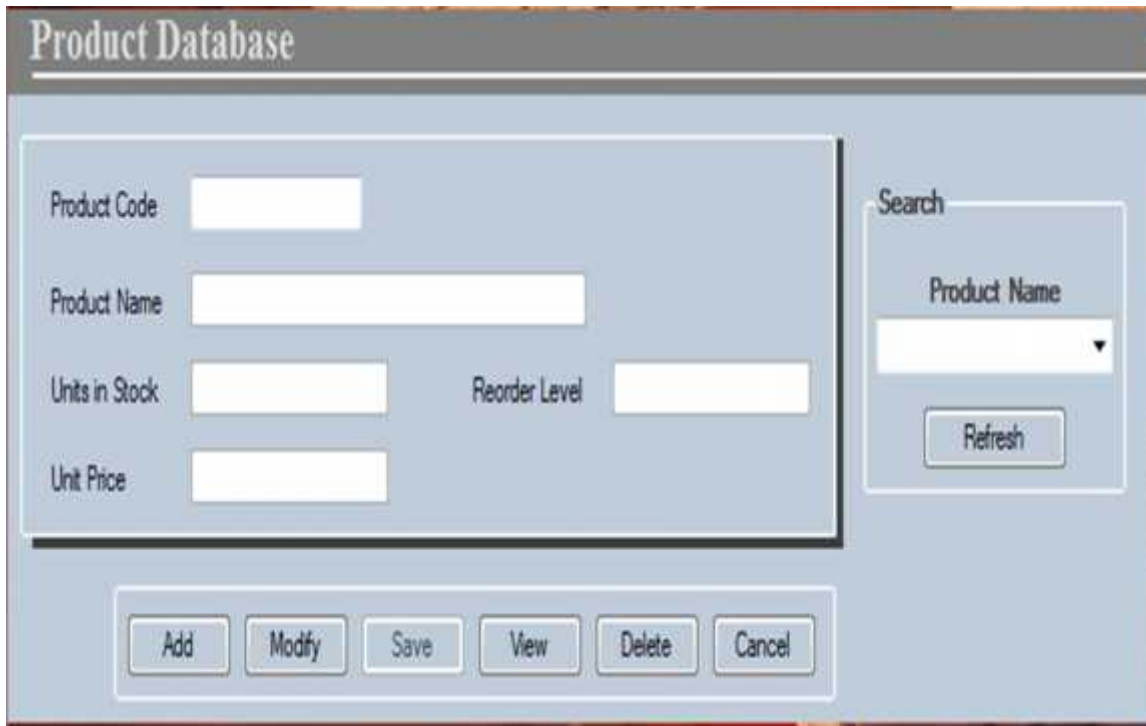
The image shows a software interface titled "Product Database". It features a central form with several input fields: "Product Code", "Product Name", "Units in Stock", "Unit Price", and "Reorder Level". To the right of the main form is a search section with a "Search" label, a "Product Name" dropdown menu, and a "Refresh" button. At the bottom of the interface is a row of six buttons: "Add", "Modify", "Save", "View", "Delete", and "Cancel".

Figure 4.4 Product information Form.

The product database consists of the following information about which a product will be stored in the database. Each product is given a unique product code which is followed by the product name. The unit in stock shows the number of products in stock in the database. The unit price shows the cost of the product in stock. On this form user can make a search for its existing products as shown above and can make any modification to existing product information. Reorder level of the product in stock prompts the users to inform management to order for more product.

-) The **Add** button is clicked to add information about new product.
-) The **Save** button saves information about the product into the database.
-) The **View** button helps to view a product in order to delete or modify.
-) The **Modify** button modifies the information about the product.
-) The **Delete** button deletes the information about the product from the database.

) The cancel button closes the form.

The image shows a software interface for recording bank transactions. The main window is titled "BANK TRANSACTION". On the left side, there are several input fields: "Bank" (a dropdown menu), "Date" (a date picker set to 5/22/2012), "Account" (a text box), "Current Balance" (a text box), "Amount" (a text box with the value 0), "Transaction Type" (a dropdown menu), and "By" (a text box). On the right side, there is a "Control Box" containing four buttons: "Add", "Save", "View", and "Exit".

Figure 4.5 Bank transaction form.

The figure 4.5 contains a control box which helps to add, save and view any transaction made with a bank. This form helps the administrator to choose the transaction type to make with the bank either to deposit or withdraw. It gives the actual current balance of the company without contacting the bank. Every transaction made with the bank will save here for future reference. It gives the user the right to add as many banks as needed by the company.

-) The **Add** button is clicked to add information about a bank in which the company transacts with.
-) The **Save** button saves information about the bank into the database.
-) The **View** button helps to view information about the bank.
-) The **Exit** button closes the form.

Figure 4.6 Payment information form.

Payment of transaction can be either made by cash or cheque. There is an option to choose whether the payment is from a debot or a supplier. When the payment type is by cheque there is field provides to save the cheque number in addition for future reference. Payment of transaction can be made in segment when a debtor is unable to settle their debt. The check balance box helps to check balance both the debtor and the supplier as shown in figure 4.6.

-) The **Add** button is clicked to add information about either a debtor or a supplier payment.
-) The **Save** button saves information about either a debtor or a supplier payment into the database.
-) The **Edit** button modifies the information about a debtor or a supplier.
-) The **Delete** button deletes the information about the debtor or a supplier from the database.
-) The **Exit** button closes the form

Figure 4.7 Delivery information form.

This figure 4.7 helps in updating the stocks. It actually gives the clear information from the company's suppliers. It give the delivery number, the day it was delivered followed by the suppliers name,the invoice number,the expiry date of the product.

-) The **New** button is clicked to update the stock of products from the company's suppliers.
-) The **Delivered** button saves the information and updates the product in stock.
-) The **Add invoice** button saves the invoice number, the quantity of the product and the cost of the product to be paid to the supplier.
-) The **Close** button closes the form.
-) The text box product name contains the name of the product.
-) The text box product code contains the code of the product. It will automatically appear when the check button is clicked.

The image shows a software interface for a sales transaction form. At the top left, there is a 'Transaction ID' field. To its right, under the heading 'Option', are two radio buttons: 'Customer' and 'Debtor'. Below this is a section with three input fields: 'Name' (with a dropdown arrow), 'Date', and 'Invoice'. Underneath is a 'Check Product' section with a dropdown menu and a 'Check' button. The next section contains four input fields: 'Product Code' and 'Unit Price' in the top row, and 'Product Name' and 'Units in Stock' in the bottom row. At the bottom center, there is a 'Total Amount' field displaying the value '0'. On the right side, there is a 'Control Box' containing six buttons: 'New', 'Order', 'Add Invoice', 'Print', 'Refund', and 'Close'. Above the control box is a rectangular area containing a photograph of a whole, raw chicken.

Figure 4.8 Sales transaction form.

The figure 4.8 consists of the following information about which all daily sales will be stored in the database. Each sale is given a unique code which is followed by the customer name, the date, invoice number and the product the customer want to purchase. When the product is checked the product code, the unit price the product and the unit in stock will automatically appear. And therefore gives the room for the product to be ordered. After that, the transaction made can be printed out.

-) The **New** button is clicked to order for a product for a customer.
-) The **Add** invoice helps to save an unpaid invoice of a customer by storing it in the database.
-) The **Refund** button helps to refund a product which was wrongly input into the systems.
-) The **Close** button closes the form.
-) The **Print** button is clicked to print out the receipt for the customer.
-) The **Order** button is clicked to order for a product.

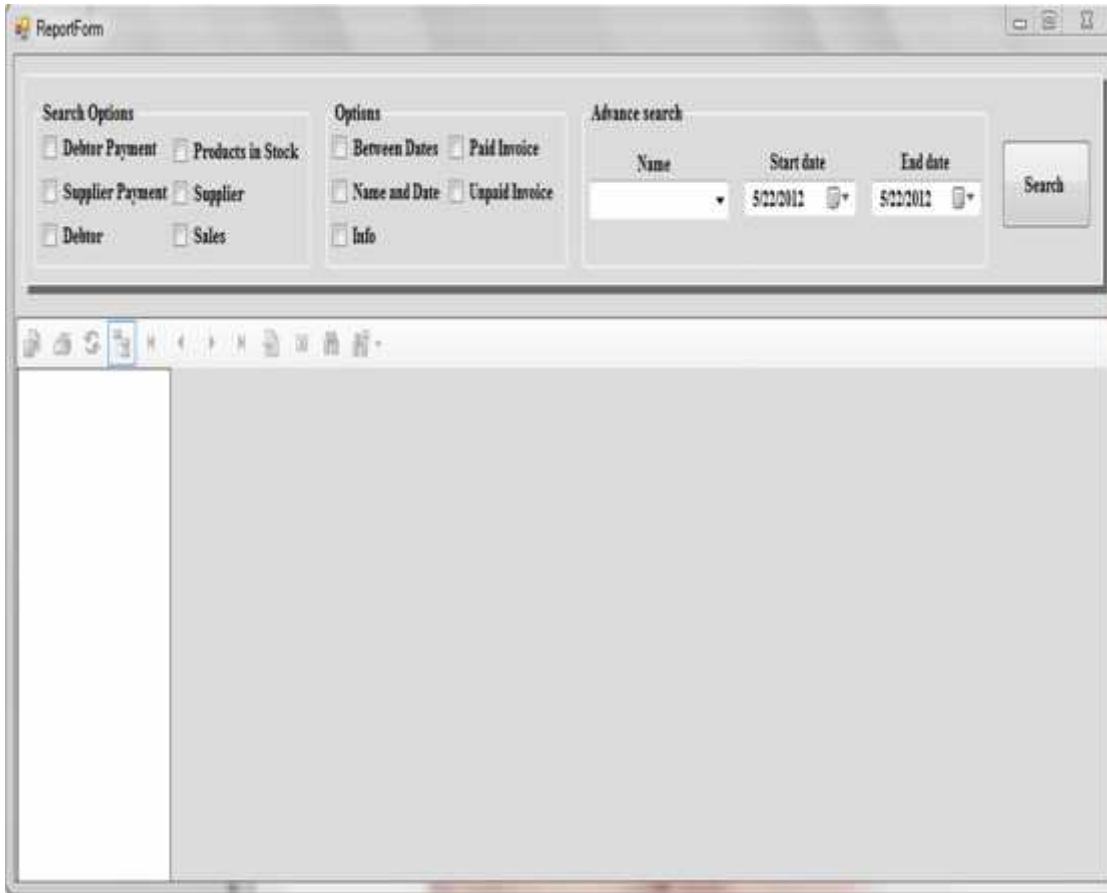


Figure 4.9 Report Information Form.

This figure 4.9 contain all detailed report which can be obtain from this project. The search option in the figure 4.9 shows the type of report to be generated either a supplier, a debtor or the daily sales transaction. After that, it gives another option to search for paid and unpaid invoice. The advance search help to search for report between a specific period of time. The check boxes helps to locate the needed report.

-) The Search option guides the administrator to select the required report to be generated.
-) It gives another option for the administrator to select another option for either supplier's payment bet.ween some specific dates.
-) The Search button is clicked to produce the specified report.

The figure 4.1.0 shows how a different user tried to use system with the administrator's username to log into the system but made a wrong entry of the password. Since the user has no privilege to log in as an administrator the invalid log in message appears on the screen.



Figure 4.1.0 Invalid Login



Figure 4.1.1 Invalid Input

The figure 4.1.1 shows a wrong entry in entering the company name of a supplier in the supplier's database. The system prompts the user that he/she has made a wrong input. The text box support only characters.

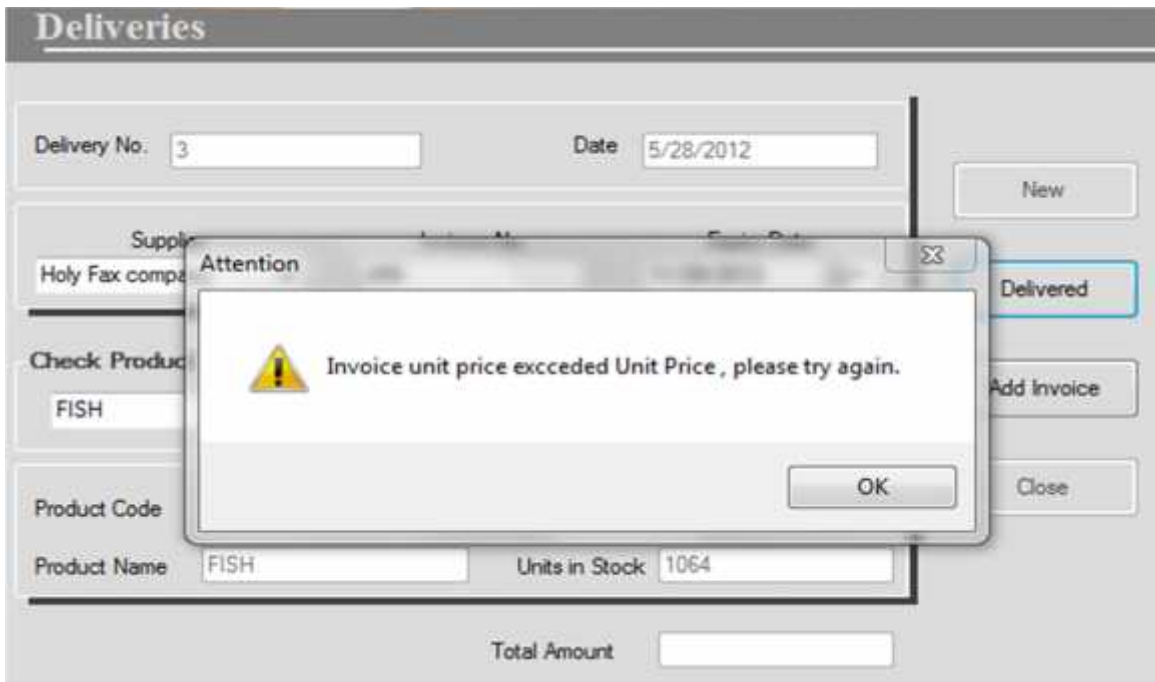


Figure 4.1.2 Unit Price Exceeded Unit Price

The delivery form help to update the stocks for a product in the database. This form has the unit price for each product in the database. When the administrator try to update the stocks and try to enter a different unit price for a product, the above message pop out as in figure 4.1.2.

4.3 EVALUATING THE PROJECT

After meeting all the requirement of users, the motive of carrying out this project has been a met. The system takes charge of most of the manual operations being performed by the users. The manual operations like producing reports about the daily sales, supplier, and debtors. Security of the system has been taken care of, with regard to users and administrator.

4.4 EVALUATING THE SOLUTION

The desired solution for the project was attained and could easily be installed by the company with little training of staff and other users.

4.5 ACCEPTANCE TESTING

The user formally accepts the system when it has successfully passed the acceptance test. The purpose of an acceptance test is to prove to the client that the system meets the business requirement agreed in the functional specification. The acceptance test is run by a client or under designer's supervision. When all problems have been resolved, the client signs for acceptance of the system. The result from this phase is a system that works to the satisfaction of the client as defined in the requirements specification document and any related change request documents.

4.6 DATA TAKE-ON AND CONVERSION

During the transition period from the old system to the new system, it is essential that the data the organization already has is safely transferred to the new system. It is unlikely that the format of the data on the two systems will be the same, as the organization may, for example, be moving from a file-based data to a database system.

CHAPTER FIVE

CONCLUSION

5.0 SUMMARY OF PROBLEMS, ACHIEVEMENTS AND RECOMMENDATIONS

When developing this software, there were many problems associated with

5.1 PROBLEMS

Developing an IT solution to manage a cold store management system such as chilling cold store management system was an overwhelming task. Getting the vital information to build up the system was a difficult task. The company was not ready to give out all the necessary information because some of their confidential documents might leak. Other problems faced were learning Microsoft Visual Basic.Net, Microsoft SQL Server 2005 and Crystal Reports for generating various reports.

5.2 ACHIEVEMENTS

For most solutions, choosing the appropriate tool will require the developer to weigh the importance of several solution criteria. While there are some many solutions that might have accomplished this task. Microsoft Visual Basic.Net, Microsoft SQL Server 2005 has made it simple and easy by meeting the requirements. The project was able to automate the transactions in Chilling Company limited through the processes of helping the company to move from the process paper booking to a computerized base.

5.3 RECOMMENDATION

To improve this software, Developers should include more security measures such encryption of data in storage. Although time limitation did not permit the addition of some useful features which might be needed by the company, yet it is hoped that the process of maintenance will meet this need.

REFERENCES

1. AASHTO. (1997) 21st Century Asset Management. American Association of State Highway and Transportation Officials, Washington, D.C.
2. Dalton, Patrick. (2000) Microsoft SQL Server Black Book, Coriolis Group Books Inc.
3. James A. O'Brien and George M Marakas (2008) Management Information System, 7th edition, The McGraw Hill Companies.
4. Kenneth C. Laudon and Jane. P. Laudon, Prentice Hall Pvt (2006) Management Information Systems, managing the Digital Firm, 8th edition.
5. Pankaj Jalote, Narosa, N-Delhi.(2005) An integrated approach to software engineering,
6. Peter Norton, Introduction to computers, 6th edition, The McGraw Hill Companies.
7. Pressman, Roger S. Software Engineering(2010) A Practitioner's Approach. 3rd Edition. McGraw-Hill.
8. Sabharwal Sangeeta (1998) Software Engineering
9. Senn, James A. (1989) Analysis and Design of Information Systems. McGraw-Hill, 2nd Edition.
10. Nemmers, C. "Transportation Asset Management." Public Roads Magazine. Data accessed at www.tfhr.gov/pubrds/july97/tam.htm (3rd April,2012).
11. OECD. (2001) Asset Management for the Road Sector. Organization for Economic Co-Operation and Development, Paris.

APPENDIX

CODES

CODES ON THE MAIN FORM

```
Public Class Form1

    Private Sub btnsupplierinfo_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsupplierinfo.Click
        Dim SUP As New Supplier_Database_Form2
        SUP.Show()

    End Sub

    Private Sub btnsuppliertransaction_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPaymenttransaction.Click
        Dim SUPP1 As New Payment_Transaction_Form2
        SUPP1.Show()

    End Sub

    Private Sub _Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnProduct.Click
        Dim PRO As New Product_Database_Form2
        PRO.Show()

    End Sub

    Private Sub btnrevenue_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDebtor.Click
        Dim CUS As New Debtor
        CUS.Show()

    End Sub

    Private Sub btndebtors_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnBank.Click
        Dim BANK As New Bank_Transaction_Form2
        BANK.Show()

    End Sub

    Private Sub btnReport_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDeliveries.Click
        Dim DILI As New Deliveries_Form2
        DILI.Show()

    End Sub

End Class
```



```

Private Sub btnexit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnexit.Click
    Me.Close()

End Sub

Private Sub btnSales_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSales.Click
    Sales_TransactionForm2.Show()

End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick

    Static x

    If x = 0 Then
        PictureBox1.Visible = True
        PictureBox2.Visible = False
        PictureBox3.Visible = False
        PictureBox4.Visible = False
        PictureBox5.Visible = False
        PictureBox6.Visible = False

        x = 1
    ElseIf x = 1 Then
        PictureBox1.Visible = False
        PictureBox2.Visible = True
        PictureBox3.Visible = False
        PictureBox4.Visible = False
        PictureBox5.Visible = False
        PictureBox6.Visible = False

        x = 2
    ElseIf x = 2 Then
        PictureBox1.Visible = False
        PictureBox2.Visible = False
        PictureBox3.Visible = True
        PictureBox4.Visible = False
        PictureBox5.Visible = False
        PictureBox6.Visible = False

        x = 3
    ElseIf x = 3 Then
        PictureBox1.Visible = False
        PictureBox2.Visible = False
        PictureBox3.Visible = False
        PictureBox4.Visible = True
        PictureBox5.Visible = False
        PictureBox6.Visible = False

        x = 4
    ElseIf x = 4 Then

```

```

        PictureBox1.Visible = False
        PictureBox2.Visible = False
        PictureBox3.Visible = False
        PictureBox4.Visible = False
        PictureBox5.Visible = True
        PictureBox6.Visible = False

        x = 5

    ElseIf x = 5 Then
        PictureBox1.Visible = False
        PictureBox2.Visible = False
        PictureBox3.Visible = False
        PictureBox4.Visible = False
        PictureBox5.Visible = False
        PictureBox6.Visible = True
        x = 0
    End If

End Sub

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    'C:\Users\Tachie\Documents\Visual Studio
2008\Projects\Inventory_System\Inventory_System\InvoiceCrystalReport2.rpt
End Sub

Private Sub btnPrint_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnPrint.Click
    Dim print As New ReportForm
    print.ShowDialog()

End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim ADD As New AddUserForm2
    ADD.Show()
End Sub

Private Sub Label13_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)

End Sub
End Class

Public Class AddUserForm2
    Public sConn As SqlConnection
    Dim SDS As DataSet = New DataSet
    Dim SDA As SqlDataAdapter = New SqlDataAdapter
    Dim SDS1 As DataSet = New DataSet
    Dim SDA1 As SqlDataAdapter = New SqlDataAdapter
    Private cmb As New SqlCommandBuilder(SDA)
    Dim SDR As DataRow

```

```

Dim SDR1 As DataRow
Dim dt1 As New DataTable
Dim dt As New DataTable

```

CODES ON ADD USER FORM

```

Private Sub AddUserForm2_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    sConn = New SqlConnection("Data Source=BIZZLE-PC\SQLEXPRESS;Initial
Catalog=Inventory;Integrated Security=True")

```

```

    Dim SDA1 As New SqlDataAdapter("SELECT* FROM PRIVILEGES", sConn)

```

```

    Dim dt As New DataTable
    SDA1.Fill(dt)

```

```

    cboAccessRight.DataSource = New BindingSource(dt, Nothing)
    cboAccessRight.ValueMember = "Name"

```

```

    cboAccessRight.ResetText()

```

End Sub

```

Private Sub cboAccessRight_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cboAccessRight.SelectedIndexChanged

```

```

    sConn.Open()

```

```

    SDA.SelectCommand = New SqlCommand("select * from PRIVILEGES where
Name=' " & Me.cboAccessRight.Text & "'", sConn)
    SDS.Clear()
    SDA.Fill(SDS, "PRIVILEGES")

```

```

    If SDS.Tables(0).Rows.Count > 0 Then
        SDR = SDS.Tables(0).Rows(0)

```

```

        txtID.Text = SDR("PRIVILEGE_ID")

```

End If

```

    sConn.Close()

```

End Sub

```

Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSave.Click

```

```

    sConn.Open()
    Dim strSQL As String

```

```

    strSQL = "INSERT INTO UserLogin(Username,Password,PRIVILEGE_ID)" & _
"VALUES('" & txtUsername.Text & "','" & txtPassword.Text.ToUpper.Trim
& "','" & txtID.Text & "'"

```

```

    Dim SCmd As SqlCommand = New SqlCommand(strSQL, sConn)

```

```

        SCmd.ExecuteNonQuery()
        SCmd.Dispose()
        MsgBox("saved user")

        sConn.Close()
    End Sub

    Private Sub btnDelete_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDelete.Click
        sConn.Open()

        Dim a
        a = InputBox("Enter Username to delete")

        Dim strSQL As String

        strSQL = "DELETE FROM UserLogin WHERE Username = '" & a & "'"

        Dim SCmd As SqlCommand = New SqlCommand(strSQL, sConn)
        SCmd.ExecuteNonQuery()
        SCmd.Dispose()

        MsgBox("Record has been successfully deleted.", vbInformation,
"Confirm")
        sConn.Close()
    End Sub

    Private Sub txtUsername_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtUsername.KeyPress
        Dim ch As Char = e.KeyChar
        If Char.IsDigit(ch) Then 'Ristricting TxtName To Input Only
Characters
            e.Handled = True
            MsgBox("Invaild input.Input Only Characters",
MsgBoxStyle.Information)

        End If
    End Sub

End Class

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    'Label13.Left = Label13.Left + 3
    'If Label13.Left > Me.Width Then Label13.Left = 0

    End Sub

    Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAdd.Click
        cboBankName.ResetText()
        txtAccount.Clear()
        txtAmount.Clear()
        txtBy.Clear()

```

```

txtCurrentBalance.Clear()
cbotransactionType.ResetText()
dtpDate.Text = Now.Date

cboBankName.Enabled = True
txtAccount.Enabled = False
txtAmount.Enabled = True
txtBy.Enabled = True
txtCurrentBalance.Enabled = False
cbotransactionType.Enabled = True
dtpDate.Enabled = True

btnSave.Enabled = True
btnAdd.Enabled = False

```

End Sub

```

Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSave.Click

```

```

    sConn.Open()
    Dim strSQL As String

    strSQL = "INSERT INTO
Bank_Transaction(Bank,Date,Account,CurrentBalance,Amount,TransactionType,Person)" & _
"VALUES('" & cboBankName.Text & "','" & dtpDate.Text & "','" &
txtAccount.Text & "','" & txtCurrentBalance.Text & "','" & txtAmount.Text &
"',''" & cbotransactionType.Text & "','" & txtBy.Text & "')"
    Dim SCmd As SqlCommand = New SqlCommand(strSQL, sConn)
    SCmd.ExecuteNonQuery()
    SCmd.Dispose()

    sConn.Close()

    MsgBox("Record successfully saved.", vbInformation, "Save")

    btnSave.Enabled = False
    btnAdd.Enabled = True

```

End Sub

```

Private Sub Bank_Transaction_Form2_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
    sConn = New SqlConnection("Data Source=BIZZLE-PC\SQLEXPRESS;Initial
Catalog=Inventory;Integrated Security=True")

```

```

    Dim SDA1 As New SqlDataAdapter("SELECT* FROM Bank_Setup", sConn)

    Dim dt As New DataTable
    SDA1.Fill(dt)

    cboBankName.DataSource = New BindingSource(dt, Nothing)
    cboBankName.ValueMember = "BankName"

```

```

        cboBankName.ResetText()
        txtAccount.Clear()
        txtCurrentBalance.Clear()
    End Sub

    Private Sub cboBankName_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cboBankName.SelectedIndexChanged

        SDA1.SelectCommand = New SqlCommand("select * from Bank_Setup where
BankName='" & Me.cboBankName.Text & "'", sConn)
        SDS1.Clear()
        SDA1.Fill(SDS1, "Bank_Setup")

        If SDS1.Tables(0).Rows.Count > 0 Then
            SDR1 = SDS1.Tables(0).Rows(0)
            txtAccount.Text = SDR1("AccountNo")
            txtCurrentBalance.Text = SDR1("CurrentAmount")
        End If
    End Sub

    Private Sub cbotransactionType_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cbotransactionType.SelectedIndexChanged
        If cbotransactionType.Text = "Deposit" Then

            txtCurrentBalance.Text = Val(txtCurrentBalance.Text) +
Val(txtAmount.Text)

            ElseIf cbotransactionType.Text = "Withdrawal" Then

                txtCurrentBalance.Text = Val(txtCurrentBalance.Text) -
Val(txtAmount.Text)

            End If
        End Sub

    Private Sub btnEdit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)

        End Sub

    Private Sub btnView_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnView.Click
        Dim view As New BankViewForm2
        view.ShowDialog()

    End Sub

    Private Sub txtAccount_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtAccount.KeyPress
        Dim ch As Char = e.KeyChar
        If Char.IsLetter(ch) Then 'Ristricting CNIC To Input Only
Digits(any number)

```

```

        e.Handled = True
        MsgBox("Invaield input.Enter Digit(Number)",
MsgBoxStyle.Information)
    End If
End Sub

Private Sub txtCurrentBalance_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtCurrentBalance.KeyPress
    Dim ch As Char = e.KeyChar
    If Char.IsLetter(ch) Then 'Ristricting CNIC To Input Only
Digits(any number)
        e.Handled = True
        MsgBox("Invaield input.Enter Digit(Number)",
MsgBoxStyle.Information)
    End If
End Sub

Private Sub txtAmount_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtAmount.KeyPress
    Dim ch As Char = e.KeyChar
    If Char.IsLetter(ch) Then 'Ristricting CNIC To Input Only
Digits(any number)
        e.Handled = True
        MsgBox("Invaield input.Enter Digit(Number)",
MsgBoxStyle.Information)
    End If
End Sub

Private Sub txtAmount_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles txtAmount.TextChanged

End Sub

Private Sub txtAccount_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles txtAccount.TextChanged

End Sub

Private Sub txtCurrentBalance_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles txtCurrentBalance.TextChanged

End Sub

Private Sub txtBy_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtBy.KeyPress
    Dim ch As Char = e.KeyChar
    If Char.IsDigit(ch) Then 'Ristricting TxtName To Input Only
Characters
        e.Handled = True
        MsgBox("Invaield input.Input Only Characters",
MsgBoxStyle.Information)

    End If
End Sub

```

```
Private Sub txtBy_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles txtBy.TextChanged

End Sub

Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnExit.Click
    Me.Close()

End Sub
End Class
```