DATA MINING FOR INVENTORY CONTROL FOR BLESSED SUPERSTORES LIMITED: A PILOT STUDY

By

Chukwu Jane Ngozi (B.Eng)
20074616268

A THESIS SUBMITTED TO THE POSTGRADUATE SCHOOL
FEDERAL UNIVERSITY OF TECHNOLOGY, OWERRI.

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE AWARD OF MASTER OF ENGINEERING, M. Eng
IN
COMPUTER ENGINEERING

DECEMBER, 2010

CERTIFICATION

This is to certify that **Chukwu Jane Ngozi**, a postgraduate student with registration number **20074616268**, has satisfactorily completed the requirements for the award of Master of Engineering (M.Eng) in Electrical/Electronics Engineering (Computer Engineering).

Engr. Dr. F.K. Opara ………………………

(Supervisor) Signature/Date

Engr. Dr. E.N.C. Okafor ……………………….

(HOD, E/E Dept.) Signature/Date

Engr.Prof.C.D.Okoronkwo ………………………

(Dean PGS) Signature/Date

Engr.Prof.C.Osuagwu ……………………….

(External Examiner) Signature/Date

DEDICATION

This project work is dedicated to the Almighty God, the creator of the whole universe, and the source of all wisdom; and to my loving family, for all their love and support.

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to my supervisor, Engr. Dr. F.K. Opara, for his guidance, suggestions, and corrections, which all contributed to the success of this project work. I also acknowledge the wonderful contributions of the following people; my HOD, Engr. Dr. E.N.C. Okafor, the Dean, School of Engineering and Engineering Technology, Engr. Prof. O. N Oguoma, Engr. Prof. S. O. E. Ogbogu, Engr. Mrs. Eze, and my entire course mates.

My profound gratitude equally extends to my loving family, for the moral and financial support they gave me throughout the period of my academic programme. I would also like to express my special thanks to Engr. O. Ude, and others, who contributed immensely to the completion of this work.

Finally, my greatest thanks go to the Almighty God for His unceasing mercies and abundant blessings which He continually showers on me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

As the nation’s economy has tightened, managers of retail businesses have been greatly challenged in recent years to become more strategic in their planning. These managers need fast, reliable and interpretable analysis of different transaction data in order to effectively assess the performance of their business organizations. The introduction of computerized Point-of-Sale terminals has made it possible for retailers to easily accumulate large stores of sales data. This wealth of data provides the required knowledge for competitive advantage, through the application of data mining. Among other benefits, data mining can be used to enhance inventory control. With data mining, retailers are able to conduct detailed analysis of sales transactions to accurately forecast demands that match customer profiles. The purpose of this thesis is to explain how the data mining technology can be used in small retail firms to significantly increase sales and improve inventory control. The Blessed Superstores Limited, a small retail outfit, is used as a pilot study. As a necessary first step to data mining, a computerized Point-of Sale system, which is to be used to collect all transaction data at the superstore, has been developed. The data mining process would only commence after sufficient data has accumulated in the database.

CHAPTER 1

1.0 INTRODUCTION

1.1 Overview of the Project

Traditionally, managers have always performed the task of extracting useful information from recorded data; the exponential growth in the amount of data being collected in modern business has however necessitated the use of computers. As data sets have grown in size and complexity, a shift away from direct hands–on data analysis towards indirect automatic data analysis has become necessary. Managers often struggle to find answers to questions such as:

- What kind of marketing strategy is most likely to increase sales?
- What can customer-purchasing pattern reveal about improving inventory control?
- What is the most effective way to manage customer relations to increase revenue? [1]

With the improvements in technology, and research in machine learning, strategic decision making by these managers can be effectively carried out by the use of data mining [2].

Data mining is the process of sorting through large amounts of data to pick out relevant information [3]. In relation to enterprise resource planning, it is the statistical logical analysis of a large set of transaction data, looking for patterns and trends that can aid decision making [4]. Currently being utilized by big stores as the Federated Department Stores, Dun and BradStreet, data mining has been successfully deployed in businesses to reduce costs, and increase sales. It is being used in all phases of Customer Relationship Management (CRM), including getting new customers, increasing revenue from existing customers, and retaining good customers [5].

Blessed Superstores Limited, an organization that operates a supermarket, wishes to improve on its business process by deploying a better inventory management plan, and identifying new merchandise opportunities. Management cannot easily implement such decisions using the available sales records. This is because the existing system cannot provide sufficient information on customer transactions.

Based on the above difficulty, the management has decided that a data mining system that will efficiently analyze customer purchase data, and extract the required information, be developed.

1.2 Objectives

The objectives of the project are as follows:

1. To develop Sales, Purchase and Inventory modules, as part of a Point of Sale (POS) system that extracts transaction data from each item that is purchased in the superstore.
2. To build a database system that will be used to store and manage the data.
3. To illustrate how the data can be analyzed, and applied to improve sales, using suitable data mining software.
4. To present the data in a useful format that will aid the management in decision making.

1.3 Significance of the Study

The transaction data of any organization usually contains lots of valuable information, such as trends and patterns, which could be used to improve its business decisions and optimize success. The volume of data generated in today's businesses, however makes it almost impossible to manually analyze them for valuable decision-making information.

This need for automated extraction of useful knowledge from huge amount of data led to the introduction of data mining tools. These tools have the capability to analyze the raw data and present the extracted high level information to the decision maker. An organization can thus apply data mining by using the knowledge about its customers implicit in a data warehouse, to reduce cost and improve its sales process.

- Blessed Superstores, a sales organization, has been experiencing serious financial losses and may in fact gradually become bankrupt.
- Because of the inefficiency of its manual systems, the organization can no longer manage its inventory properly, and most of the times, it runs out of stock. This is very dangerous in business. Maintaining proper inventory levels is critical to the success of any business organization.
- Management does not have access to the relevant information for an effective inventory control plan. The introduction of a data mining system that will efficiently analyze all point of sale data, to provide the required information, will greatly overcome these shortcomings.
- Even though the deployment of the proposed system may not yield the expected financial gains initially, its benefits will be justified in the long run.

1.4 Scope

The project is concerned with the design of a data mining system that will efficiently analyze all transaction data at Blessed Superstores. The required information is to be obtained by identifying the patterns, or relationships among all these data. To this end, the project shall study all the sales records in the organization, during the specified period.

The project shall aim towards designing a total system, but only three modules, namely: - Sales, Inventory and Purchase shall be implemented because of time and financial constraints. Considering the size of the required data which is expected not to exceed 50 gigabytes, a simple Relational Database Storage System (RDBS) shall be used in processing and managing the data. The mining process can only commence after sufficient data have been collated. As a result, the project shall only explain how the data can be analyzed, using data mining tools, to uncover customer trends and patterns that would be used to successfully manage inventory and make other business decisions.

1.5 Thesis Organization

The project is organized as follows:

Chapter 1 deals with the introduction. An overview of the project is presented, followed by brief highlights of its objectives, significance and scope. In chapter 2, a detailed review of relevant information on the data mining process is presented. Chapter 3 discusses the project methodology, illustrating both the old and new system designs, as well as the development steps for the project.

Chapter 4 explains how the POS system can be successfully implemented, with a description of the various tests that were carried out to ascertain the workability of the program. The project is thereafter concluded in Chapter 5. The Visual Basic source codes for the implementation of the Point- of- Sale (POS) system will be presented in appendix E of this project.

CHAPTER 2

2.0 LITERATURE REVIEW

2.1 Data, Information and Knowledge

Data is the term used to describe the basic facts about the activities of a business [6]. It could also refer to any facts, numbers or text that can be processed by a computer. The patterns, associations or relationships among different data can provide information. For example, analysis of retail point of sale transaction data can yield information on which products are selling, and when.

By definition, information is data that have been processed in such a way as to be useful to the recipient [7]. In every system, the users receive the information in the form of output. Information can be converted into knowledge about historical patterns and future trends. For example, summary information on a supermarket can be analyzed to provide knowledge that can be used for promotional activities.

2.2 The Concept of a System

A system is usually defined as a set of interacting elements which come together to achieve a specific objective [8]. Data processing systems are concerned with the elements, which inter relate in a systematic way to process data into information. These elements will consist of procedures, rules, files, computers, pens, papers, and above all, human beings. The way in which this interaction occurs is usually of interest to the system analyst, whose duty it is to see whether improvements can be made by reorganizing the procedures or introducing new equipment.

2.2.1 The System Identification

Systems are usually identified by the elements of which they are composed. These elements delineate the boundary of the system's environment, which usually affects the way the system operates.

Relationships between systems are called interfaces, and each subsystem may interface with every other subsystem within the system. The interface normally takes the form of the passage of data or information among the subsystems. Diagrammatically, these concepts can be represented as shown in fig 2.1.Each of the subsystems will have a boundary, identified by its elements, such as people, files etc, and there is an interface between the subsystems in the form of information, which passes between them. For example, the sales subsystem will tell purchase subsystem how much it intends to sell a product. The purchase subsystem will then tell the sales subsystem how much it bought the product. The inventory subsystem will tell each of the other two when the product runs out of stock. Each subsystem is thus influenced by the other subsystems which form its environment (e.g. Customers, Suppliers general public, etc).

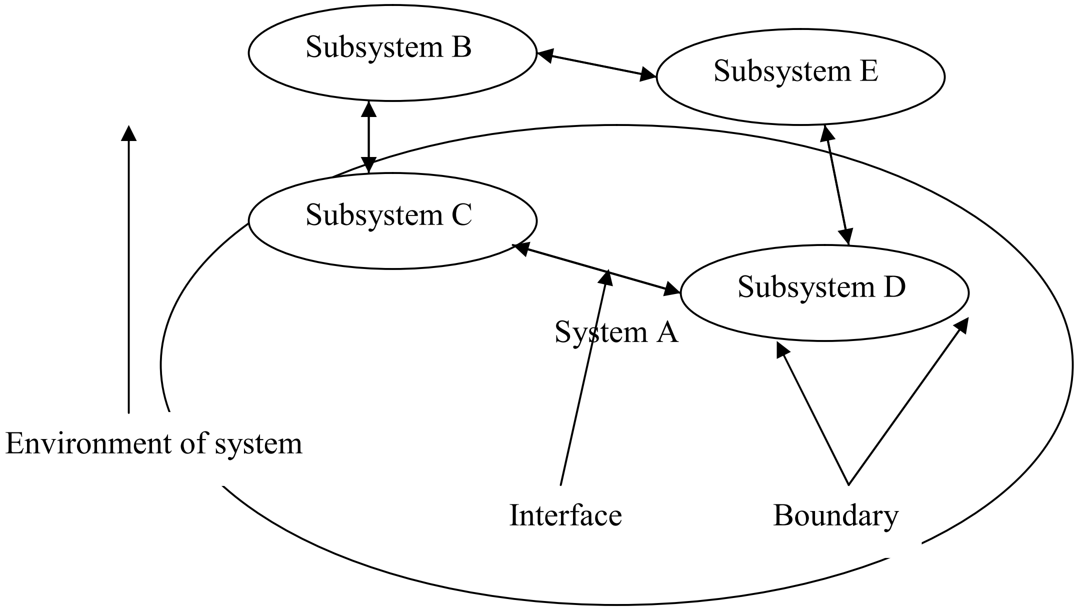


Figure 2.1 System, Subsystems, Boundary, Environment and Interface

2.3 Data Mining Systems

Data bases today contain so much data, within which lies hidden information of strategic importance. These databases are however useless if business organizations do not have the proper applications for accessing and using the data. Data mining systems, which are mainly concerned with turning data into information, allow users to analyze large databases to solve business decision problems.

2.3.1 What is Data Mining?

There are abundant definitions of data mining in the literature. From the business point of view, it is the process of identifying potentially useful patterns in data to make crucial business decisions [9]. Data mining has also been referred to as a statistical process of analyzing data stored in a data warehouse [10]. A data warehouse is an extensive data repository consisting of information from all facilities of organizations' operations, including external sources, which are maintained to support decision–making. Data within the warehouse is manipulated to create easy access by data mining tools. Data mining is mainly concerned with the analysis of data and the use of software techniques for finding patterns and regularities in large sets of data. The computer system seeks to find the patterns by identifying the underlying rules and features in the data. The analysis process starts with a set of data, uses a methodology to develop an optimal representation of the structure of the data during which time knowledge is acquired. Once knowledge has been acquired, this can be extended to large sets of data, working on the assumption that the larger data set has a structure similar to the sample data.

Data mining got its start in what is now known as "customer relationship management" (CRM), which aims to build one-to-one relationships with customers by noting their needs, remembering their preferences, and learning from past interactions how to serve them better in the future. The customer focused enterprise regards every record of an interaction with a client as a learning opportunity. The data is gathered, organized in a consistent and useful way and then analyzed to provide the required information. The data mining process can be defined by using the following six basic steps [11]:

1. Define business problem

The objectives of the data mining project are clearly stated by analyzing the business requirements. A data availability study is then carried out to ensure the available data supports the needs of the users.

2. Prepare Data

A subset of the data is selected, cleansed (by removing inconsistencies, such as flawed or missing entries), and then loaded into the data mining database.

3. Explore Data

The prepared data is explored to identify the most important fields or derived values which are to be used for predictions. Exploration techniques include calculating the minimum and maximum values, mean and standard deviation.

4. Build Models

The prepared data is separated into training and testing datasets. The training dataset is used for building the model, while the testing dataset is used to test the accuracy of the model. The structure of the mining model is defined, processed and then trained by populating it with patterns that describe the model.

5. Evaluate and Validate Model

The results generated from the model are evaluated to determine its accuracy rate. The model is then tested in the real world by trying it out on a small collection of transaction data.

6. Deploying the Model

The model is deployed for use and can be incorporated into an application such as a POS system, which serves as an inventory ordering system that automatically generates an order when the forecast inventory levels drop below a threshold.

2.3.2 Data Mining Goals

Most data mining goals fall under the following categories:

- Classification (examining the feature of a specific data item, and assigning it to one of a predefined set of classes).
- Estimation (given some input data, coming up with a value for some unknown continuous variable such as income or credit card balance).

- Prediction (records are classified according to some predicted future behavior or estimated future value).
- Association rules or affinity grouping (determine which things/data items go together and their relationships).
- Clustering (segmenting a population into a number of subgroups or clusters).
- Feature Extraction (creates new attributes or features using linear combinations of the original attributes).

Each data mining goal or function specifies a class of problems that can be solved. Data mining functions fall generally into two categories: Supervised and Unsupervised, derived from the science of Machine Learning. The first three tasks- classification, estimation and prediction- are all examples of directed knowledge discovery (supervised learning). In supervised learning, the goal is to use the available data to build a model that describes some particular variable of interest, such as income or response, in terms of the rest of the available data ("class prediction"). The building of a supervised model involves training, a process whereby the software analyzes many cases where the target value is already known. In the training process, the model "learns" the logic for making the predictions. For example, a model that seeks to identify the customers who are likely to respond to a promotion must be trained by analyzing the characteristics of many customers who are known to have responded or not responded to a promotion in the past.

The next three tasks – affinity grouping or association rules, clustering, and feature extraction - are examples of undirected knowledge discovery (unsupervised learning). In unsupervised learning, no variable is singled out as the target; the goal is to establish some relationship among all the variables ("class discovery"). Unsupervised learning attempts to find patterns or similarities among groups of records without the use of a particular target field or collection of predefined classes. There is no previously known result to guide the algorithm in building the model.

2.3.3 Data Mining Algorithms

A data mining algorithm is a specific technique or procedure for producing a data mining model. An algorithm uses a specific model representation and may support one or more mining functions. The choice of a particular combination of algorithms to apply in a particular situation depends on both the nature of the data mining task to be accomplished, and the nature of the available data.

The most commonly used algorithms in data mining are:

1. Statistical / Predictive Techniques, which use linear or logistic regression to create predictive models, based on the analysis of the dependency of some attribute values upon the values of other attributes in a given set of data items. Such models can be used to predict a response of interest.
2. Artificial Neural Networks are a non-linear predictive decision making model which uses existing data, with a known outcome, to train a model which can then be used to make predictions on data with unknown outcomes.
3. Genetic Algorithms are techniques that are based on the concept of natural evolution using genetic combination, mutation and natural selection as a form of optimization.

4. Decision Trees are tree shaped decision models that generate rules for classifying a data set. Each model represents sets of decisions .Specific decision tree methods include Classification Regression Trees (CART) and Chi Square Automatic Interaction Detection (CHAID).

5. Nearest Neighbor Method is a data classification technique based on similar records within a historical database.

6. Rule Induction is the use of if-then rules from data, based on statistical significance.

7. Data visualization is the pictorial representation of relationships among data.

2.3.4 Data Mining versus other Technologies

The type of information available to retailers with no computer for data analysis, retailers with computers using traditional data analysis capabilities and retailers with computers using data mining capabilities will be compared in this section to illustrate the value of data mining as input to strategic planning for current retailers. The following three types of questions are commonly encountered by retailers and thus serve to demonstrate realistic inputs and outcomes. Table 2.1 summarizes this comparison.

Case A - Tracking promotional effectiveness:

Question: “How well did I do in my last promotion?”

1. <u>No computer</u>- In tracking sales and promotional effectiveness with no computerized point-of-sale; a retailer would have to track information manually. Cash registers provide general information such as total sales and sales by department; however discounts and coupon sales would have to be tracked manually. Sales and discount information must be recorded in a ledger to make comparative data analysis with past promotions. Customer information would also have to be tracked manually. All these require much effort, and time.

2: Using a computer with traditional data analysis capabilities-Computerized point of sale software programs can track sales information, but retailers must formulate predetermined questions to analyze and all analyses are based on historical data that have been accumulated. Questions that retailers can answer through the use of reports include:

a) "Did my sales increase during the time of my promotion?"

b) "Which customers bought from me during the promotion?"

3: Using a computer with data mining capabilities- Databases are generated with similar sales information as the traditional data analysis software programs. The retailer however does not necessarily need to formulate predetermined questions to analyze. Data mining automatically analyses the information and can find relationships among the data to answer such questions as:

a) "Why are my discount coupons or promotions not attracting the sort of return I was expecting?"

b) "How do I get my other stores to match the sales figures of my main branch?"

Using data mining in analyzing promotion can produce information such as:

"Coca cola is purchased 65% of the time when sold at regular price and 85% of the time during promotions [12]".

Case B- Inventory Control:

Question: "How do I avoid stock outs on fast moving items?"

1. No computer- When retailers do not use computers to keep track of inventories, all Stock Keeping Unit (SKU) counts must be done by hand. This is very time consuming and carries great potential for human error. Counts must be done frequently and manually on fast moving items. "Eye balling" (visually scanning inventory) becomes a frequent practice, reducing accuracy. This type of inventory control reduces response time on reorders and increases the potential of missed

sales due to stock outs. Commonly utilized figures such as stock to sales ratios, turnover, and gross margin return on investments must be calculated by hand with sales receipts, inventory counts and a calculator. This is time consuming, subject to human error and is most likely computed for longer periods of time.

2. Using a computer with traditional data analysis capabilities- When using a computerized inventory control system, linked to a point of sale program, retailers can maintain a perpetual inventory. This is a much more accurate form of inventory control, allows for greater control of sizes and colors within styles and can free up the time spent counting by hand. Inventory reports can be generated to aid in answering predetermined questions such as:

a) “Which of my SKU’s am I running low on?”

b) “Of my top ten sellers, what are my stock levels?”

3. Using a computer with data mining capabilities- When maintaining inventory control using data mining, a retailer is not limited to sales volume and stocks levels as a means to perform analysis. Data mining can go much deeper into the database to create information such as:

a) Forecast seasonal demands that match customer profiles.

b) Forecast an increase in demand due to promotion to help inventory correctly when something goes on promotion.

c) Analyze inventory that performs differently in different market segments and location. This allows for better utilization of floor space within multiple store chains.

d) Data mining can compile a sales profile for each SKU by analyzing all possible influences, such as sales trends, seasonal patterns, demographics etc. This profile is modified overtime and allows retailers to predict just-in-time purchasing way ahead of actual sales.

Case C- Facilitating Buying Decisions:

Question: "How can I determine which new products to carry?"

1. No computer- Determining new products and styles without the aid of computerized reports can be difficult. Predicting what your customers may like without some sort of customer database can be a shot in the dark. Retailers may in this situation depend on discovering what is selling in other stores and markets, and scrambling to get their hands on stock while customer demand is still high. All too often, the goods are no longer available or the demand has dropped, and retailers find themselves over stocked, with markdowns being inevitable.

2. Using a computer with traditional data analysis capabilities- Retailers who have the benefit of sales and inventory reports generated from traditional database queries have the advantage of detailed historical information as to what products have performed well in the past, and the type of products particular customers tend to purchase. However, this information is purely historical and predicting new product success is somewhat a guessing game.

3. Using a computer with data mining capabilities- The use of data mining in making new product determinations eliminates a great deal of the guess work. It allows for greater understanding of various levels of customer sophistication to aid in determining new sales or services [13]. In deciding new products to stock, data mining can mine for:

   a) Customer complaints, and pay attention to customer feedback on what products that are currently being sold to yield new product information.

   b) Customer behavior with existing products, yielding new product information.

   c) Customer aspirations, rather than needs.

Table 2.1: Comparison of data mining versus no computer and traditional analyses

	No Computer	Traditional Analysis	Data mining
A: Tracking promotional effectiveness			
Total sales	Moderate	High	High
Comparative sales information	Low	Moderate	High
Discounts	Low	High	High
Coupon sales	Low	High	High
Customer information	Low	High	High
Tracking multiple promotions	Low	Moderate	High
Tracking add-on sales	Low	Moderate	High
Selling up to a higher price point	Low	Moderate	High
B: Inventory Management			
Tracking inventory counts	Low	High	High
Maintaining accurate counts	Low	High	High
Timely reorders	Low	Moderate	High
Avoiding stock outs	Low	Moderate	High
Calculating financial figures	Low	High	High
Managing fast moving items	Low	Moderate	High
C: Buying decisions			
Determine new products	Low	Low	Moderate
Accessing historical data	Moderate	High	High
Predicting customer preferences	Low	Moderate	High

Making timely purchases	Low	Moderate	High
Avoiding overstock	Low	Moderate	High
Reducing markdowns	Low	Moderate	High
Forecasting preferences / demands	Low	Moderate	High
Note: High = High level of quality Moderate = Moderate quality Low = Low level of quality			

2.3.5 Practical Applications of Data Mining

A range of current uses for data mining within the retail industry demonstrates its growing applications and deployment in strategic planning processes [14].

1. Category Management and Inventory Control

A retail company, Rubbermaid is using data mining technology for category management and merchandising optimization. Also, Wal-Mart, Proctor and Gamble, Coke and Pepsi have been using data mining for category management for years [15].

2. Market basket analysis

J. Crew Group Inc. has been combining click stream analysis from its website along with point–of-sale (POS) data from retail store locations to perform product affinity analysis. Their aim is to determine what clothes, shoes and accessories customers most often purchase together. The data will then be used to make complimentary product suggestions for on-line shoppers [16].

3. Customer Relationship Management (CRM) and Customer Profiles

Federated Department Stores has successfully combined customer and transaction data to identify their best customers and offer exclusive extras. ZCMI, a department store chain based in Utah is using data mining to integrate customer data with multiple other products and specific customer categories [17].

4. Additional possible retail applications

Other business objectives that data mining can address in a more effective and efficient manner than traditional data analysis techniques include:

- Analyzing stores hours and staffing needs within store management.
- Assortment planning in single and multi-store organizations, including inventory levels and control within merchandizing.
- Replenishment and reorder analysis with distribution.
- Markdown schedules and promotional timing within promotion.

Other business applications of data mining, apart from retail, include its use by credit card companies to determine which customers are most likely to respond to promotional offers. A fast food company such as Mr Biggs can equally use data mining to boost sales by offering coupons during Christmas and Easter seasons. Brokerage and investments firms can also use data mining to increase their customer base.

The above examples illustrate the ability of the data mining technology to transform raw data into workable and predictable solutions to complex business problems. Even though larger corporations have been data mining for years, smaller companies seem to have been left behind because of its prohibitive cost. However, the fact that these smaller companies could not afford or utilize data mining software has changed. Today, managers of these companies have come to realize that all the data they can collect has value if properly stored and analyzed. With the now mature technologies of data collection and storage, powerful multiprocessor computers and data mining algorithms, data mining is ready for application into the business community [18]. More and more software aimed at this market are also being developed [19].

2.4. Data Mining Tools

Data mining tools are software components and theories which allow users to extract information from data. The tools provide individuals and companies with the ability to gather large amounts of data and use them to make decisions. Research shows that there is no best data mining tool suit in the market. The best data mining tool may not be the most advanced or the one that gives the greatest accuracy in prediction.

Factors that determine the data mining tool to be used for a particular business problem include [20]:

1. Ability to access a variety of data sources.

2. Online / Offline data access, which specifies the type of database connection in use.- Online data access means that queries are run directly against the database and may run concurrently with other transactions. Offline data access is performed with a snapshot of the data source.

3. The underlying data model, whether relational, or one table.

4. Maximum number of tables/ rows/ attributes.

5. Database size the tool can comfortably handle.

6. Attribute types the tool can handle, either continuous or categorical.

7. Query language, which can be Server Query Language (SQL), Application Specific or through Graphical User Interface (GUI).

Generally, the choice of a data mining tool depends on application specific requirements, and considerations such as form and size of the data available, goals of the discovery process, needs and training of the end user. The advantages and disadvantages of the popular data mining tools in the market are summarized below [21].

Oracle

Advantages:

- Automatic and Custom data transformation.
- Excellent tool for data import/export, data exploration and data cleansing tasks.
- Has good variety of data mining algorithms such as the Support Vector Machine algorithm.
- Provides predictive analysis routines which manage data preparation, algorithm selection, model building and model scoring.
- Uses lift charts to evaluate response models and supports generalized linear models.
- Scoping of Nested data and enhanced handling of sparse data.
- Supports text mining by accepting text (unstructured data) attributes as input.

Disadvantage:

- No scripting interface for coding of complex problems.

SPSS Clementine

Advantages:

- Has good variety of data mining algorithms.
- Good optimal parameter search routines are built into many of the data mining algorithms.
- Power meta-learning models can be built in which the results of one modeling algorithm can be easily streamed as input to another modeling algorithm.
- Powerful (but proprietary) internal scripting language (CLEM), for creating variable processing.
- Moderately easy to use.

Disadvantages:

- Relatively little descriptive statistical or parametric statistical analysis capabilities are available directly in the tool.
- Relatively poor descriptive or output graphics forms.
- Model export for scoring outside the tool suite must be done via an optional publisher product.

STATISTICA Data Miner

Advantages:

- Provides the richest combination of parametric statistical and machine learning data mining algorithms.
- Relatively easy to use graphical programming user interface.
- Highly flexible tools for model output.

- Very scalable.
- Powerful customization options based on the standard Visual Basic language.

Disadvantages:

- Lift charts are not easily available for evaluation of neural net models.
- Training in statistical analysis is best for properly interpreting the results of the parametric statistical algorithms.

KXEN

Advantages:

- KXEN is one of the most accurate data mining tools in the market.
- Various combinations and transforms of existing variables are automatically created and included in the analysis as derived predictor variables.
- The KXEN tool is almost fully automatic. Popularly known as the "knowledge extraction engine", it can easily integrate into your data processing stream for use as a data mining engine.

Disadvantages:

- A clean data set must be submitted in the consistent coder of KXEN in the form of one record per entity to be modeled.
- There are no data preparation tools to assist input the data in this form.

Insight Miner

Advantages:

- Excellent tools for data import/ export, data exploration and data cleansing tasks and reduction of dimensionality prior to modeling.
- It is relatively easy to use by non data miners.
- Relatively cheap, and has a complete general purpose data mining suite.

Disadvantages:

- Relatively low level of automation.
- No scripting interface for coding of complex problems.
- No model exporting capabilities.

Affinium Model

Advantages:

- The data is imported into an internal spreadsheet, like STAISTICA Data Miner, but the only manipulation of the data is permitted through the edit button.
- New variables can be derived in the spreadsheet with a rich set of macro functions.
- Interpretation of the model results is very intuitive.

Disadvantages:

- No data exploration tools.
- The biggest potential draw back with this product is the almost complete lack of data preparation functions. Input data must be properly prepared in other tools before being imported into Affinium Model.

There are many data mining tools in the market, however, they are best designed for particular business problems. There is no best all-purpose data mining tool, and a data mining suite is usually designed according to the data it is mining [22].

2.5 Review of the Oracle Data Mining Tool

Oracle Data Mining (ODM) is an option to Oracle Corporation's Relational Database Management System (RDBMS) Enterprise Edition [23]. It was developed by Oracle Corporation, with its latest stable version 11gR2 released on September, 2009 for data mining and analytics.

ODM contains several data mining and data analysis algorithms for classification, prediction, regression, clustering, associations, feature selection, anomaly detection, feature extraction, and specialized analytics. It implements a variety of these algorithms inside the Oracle relational database; these implementations are integrated right into the Oracle database kernel and operate natively on data stored in the relational database tables. This eliminates the need for extraction or transfer of data into standalone mining servers.

Oracle Data Mining was first introduced in 2002. Its releases are named according to the corresponding Oracle database release. ODM is a logical successor of the Darwin data mining tool set developed by Thinking Machines Corporation in the mid-1990s and later distributed by Oracle after its acquisition of Thinking Machines in 1999. Most Oracle Data Mining functions accept as input one relational table or view. Flat data can be combined with transactional data through the use of nested columns, enabling mining of data involving one-to- many relationships.

ODM distinguishes numerical, categorical and unstructured (text) attributes. The product also provides utilities for data preparation steps prior to model building such as discretization, normalization and binning (sorting). Oracle Data Mining includes programmatic interfaces for Server Query Language (SQL), PL/SQL, and Java. ODM can be accessed using Oracle Data Miner, a Graphical User Interface "client" that provides access to the data mining functions. Oracle Data Miner provides wizards that guide you through the data preparation, data mining, data evaluation and model scoring process. GUI also allows the automatic generation of PL/SQL codes for the data mining activities. There is also an independent interface; the spread sheet Add-In for Predictive Analytics, which enables access to the Oracle Data Mining Predictive Analytics PL/SQL package from Microsoft Excel. The PL/SQL package (DBMS PREDICTIVE ANALYTICS) automates the data mining process by capturing the entire process in simple routines. Knowledge of data mining is not usually needed in order to apply Predictive Analytics. You do not need to create or use mining models. These models are trained and tested and then used to generate the results returned to the user.

Data mining with Oracle Data base

As previously stated, Oracle Database 11g automates the data mining process by using Predictive Analytics. Without user intervention, Predictive Analytics routines manage data preparation, algorithm selection, model building and model scoring. Mining functions supported by oracle data mining include; regression, classification, anomaly detection, clustering, association, feature selection and extraction. It also supports the following mining algorithms; Apriori, Decision Tree, Generalized Linear Models, K- Means, Naïve Bayes and O-Cluster.

The data mining function of association can be used to discover the probability of co-occurrence of items in a collection. The relationships between co-occurring items are expressed as association rules. Association rules are often used to analyze sales transactions. For example, it might be noted that customers who buy cereals at the superstore often buy milk at the same time. In fact, association analysis might find that 85% of the transactions that include cereals also include milk. This relationship could be formulated as the following rule.

Cereals implies Milk with 85% Confidence.

This application of association modeling is called market-basket analysis. It is valuable for direct marketing, sales promotion and for discovering business trends. Unlike other data mining functions, association is transaction based. In transaction processing, a case or a record (defined as all the data collected about a specific transaction) consists of a transaction such as the market basket. The collection of items in the transaction is an attribute of the transaction. Other attributes might be the date, time, location or user ID associated with the transaction. The collection of items in the transaction is a multi-record attribute. Transactional data is said to be in multi-record format, in which each case in the data is stored as multiple records in a table with different columns. An example of the multi-record case is shown in table 2.2.

Table 2.2: Transactional data

Case ID	Attribute
TRANS_ ID	ITEM_ ID
11	B
11	D
11	E
12	A
12	B
12	C
12	E
13	B
13	C
13	D
13	E

Since Oracle Data mining requires single-record case format (where each case in the data is stored as one record or row in a table.), the column that holds the collection must be transformed to a nested table prior to mining for association rules. Transactional data in a single-record case format is shown in table 2.3.

Table 2.3: Transactional data transformed for data mining

Case ID	Attribute
TRANS_ ID	ITEM_ PER _TRANS
	(ATTRIBUTE_NAME,ATTRIBUTE_VALUE)
11	(B,1)
	(D,1)
	(E,1)
12	(A,1)
	(B,1)
	(C,1)
	(E,1)
13	(B,1)
	(C,1)
	(D,1)
	(E,1)

Oracle Data Mining implements collections as nested rows as shown in table 2.3 above. Each nested row specifies an item name and a value. If an item is presented in a collection, it has a non- null value. An item is uniquely identified by its name and its value. Items with the same name but different values may occur across collections. For example, if one transaction includes one gallon of milk and another includes two gallons of milk, milk-1 and milk-2 are interpreted as different items.

The first step in association analysis is the enumeration of item sets. An item set is any combination of two or more items in a transaction. The maximum number of items in an item set is user-specified. If the maximum is two, all the item pairs will be counted. If the maximum is greater than two, all the item pairs, all the item triples, and all the item combinations up to the specified maximum will be counted. The maximum number of items in an item set is specified by the ASSO_MAX_RULE_LENGHT setting, which also applies to the rules derived from the item sets. Table 2.4 shows the item sets derived from the transaction in table 2.2 assuming that ASSO_MAX_RULE_LENGHT is set to 3.

Table 2.4: Item sets

Transaction item sets.	
11	(B,D) (B,E) (D,E) (B,D,E)
12	(A,B) (A,C) (A,E) (B,C) (B,E) (C,E)
	(A,B,C) (A,B,E) (A,C,E) (B,C,E)
13	(B,C) (B,D) (B,E) (C,D) (C,E) (D,E)
	(B,C,D) (B,C,E) (B,D,E) (C,D,E)

It is usually advised that the maximum rule length be decreased if you want to decrease the build time for the model and generate simpler rules. It is also desirable to only generate rules from item sets that are well-represented in the data. **Frequent item sets** are those that occur with a minimum frequency specified by the user. The minimum frequency item set support is a user-specified percentage that limits the number of item sets used for association rules. An item set must appear in at least this percentage of all the transactions if it is to be used as a basis for rules.

The ASSO_MIN_SUPPORT setting specifies the minimum frequent item set support. It also applies to the rules derived from the frequent item sets. Table 2.5 shows the item sets from table 2.4 that are frequent item sets with support > 66%. Increasing the minimum support would decrease the build time for the model, with fewer rules generated.

Table 2.5: Frequent item sets with minimum support 67%

Frequent item set transaction support
(B.C) 2 of 3 67%
(B.D) 2 of 3 67%
(B.E) 3 of 3 100%
(C.E) 2 of 3 67%
(D.E) 2 of 3 67%
(B.C.E) 2 of 3 67%
(B.D.E) 2 of 3 67%

CALCULATING ASSOCIATION RULES

By definition, an association rule is a mining function that captures co-occurrence of items among transactions [24]. A rule basically is a conditional statement that can easily be understood by humans and easily used within a database to identify a set of records. A typical rule is an implication of the form A->B, which means that the presence of item set A implies the presence of item set B, with certain support and confidence.

Oracle Data mining uses the Apriori algorithm to calculate association rules for items in frequent item sets. The Apriori algorithm calculates rules that express probabilistic relationships between items in frequent item sets. For example, a rule derived from frequent item sets containing A, B and C might state that if A and B are included in a transaction, then C is likely to also be included. The **IF** component of an association rule is known as the **antecedent**. The **THEN** component is known as the **consequent**. The antecedent and the consequent are disjoint, having no item in common.

Rules have an association Confidence, which is the conditional probability the consequent will occur given the occurrence of antecedent. The ASSO_MIN_CONFIDENCE setting specifies the minimum confidence for the rules. The model eliminates any rules with confidence below the required minimum.

Confidence in a rule is calculated by dividing the probability of the items occurring together by the probability of the occurrence of the antecedent. With reference to table 2.5, if B (antecedent) is present, what is the chance that C (consequent) will also be present?

Question: What is the confidence for the rule "IF B, THEN C"?

Solution: All 3 transactions include B (3/3 or 100%)

Only 2 transactions include both B and C (2/3 or 67%)

Therefore, the confidence of the rule "IF B THEN C" is 67/100 or 67%

Table 2.6 shows the rules that could be derived from the frequent items in table 2.5

Table 2.6: Calculating Association Rules

Frequent Item Set	Rules	Confidence
(B,C)	(if B then C)	67%
	(if C then B)	100%
(D,B)	(if B then D)	67%
	(if D then B)	100%
(B,E)	(if B then E)	100%
	(if E then B)	100%
(C,E)	(if C then E)	100%
	(if E then C)	67%
(D,E)	(if D then E)	100%
	(if E then D)	67%
(B,C,E)	(if B and C then E)	100%
	(if B and E then C)	67%
	(if C and E then B)	100%
(B,D,E)	(if B and D then E)	100%
	(if B and E then D)	67%
	(if D and E then B)	100%

If the minimum confidence is 70%, ten rules will be generated for these frequent item sets. If the minimum confidence is 60%, sixteen rules will be generated. The minimum confidence should therefore be increased if you want to decrease the build time for the model and generate fewer rules. Confidence can be expressed in probability notation as follows:

Confidence (A implies B) = P (B/A), which is equal to P (A, B)/P (A)

Another metric for evaluating the quality of rules generated by the model is the Support. The Support of a rule indicates how frequently the items occur together. It is the ratio of transactions that include all the items in the antecedent and consequent to the number of total transactions. Support can be expressed in probability notation as: Support (A implies B) = P (A, B)

A third measure used to evaluate the quality of a rule is the Lift. Lift indicates the strength of a rule over the random co-occurrence of the antecedent and the consequent, given their individual support. It is defined as the Confidence of the combination of items divided by the Support of the consequent.

Suppose the database at the superstores contains the following information:

Total hardware-store transactions: 1,000

Number which include "mouse":50

Number which include "keyboard": 80

Number which include "computer monitor":20

Number which include "mouse" and "keyboard":15

Number which include "keyboard" and "computer monitor":10

Number which include "mouse" and "computer monitor":10

Number which include "mouse", "keyboard" and "computer monitor":5

Now we can calculate:

Support for “mouse” and “keyboard” = 1.5% (15/1000)

Support for “mouse”,” keyboard” and “computer monitor” = 0.5% (5/1000)

Confidence of “mouse” Þ “keyboard” = 30% (15/50)

Confidence of “keyboard” Þ “mouse” = 19% (15/80)

Confidence of “mouse” and “keyboard” Þ “computer monitor” = 33% (5/15)

Confidence of “computer monitor” Þ “mouse” and “keyboard” = 25% (5/20)

<u>Conclusion:</u> The likelihood that a mouse buyer will also purchase a keyboard (30%) is greater than the likelihood that someone buying keyboard will also purchase a mouse (19%). The presence of this mouse- and –keyboard association (i.e. the support is 15%) is high enough to suggest a meaningful rule.

From the example, lift is calculated as

Lift of “mouse” Þ “keyboard”: 3.75 (30%/8%)

Lift of “mouse” and “keyboard” Þ “computer monitor”: 16.5 (33%/2%)

The choice of the Oracle Data Mining tool for the project is justified since it provides a good platform for integrating data mining seamlessly with database applications. As previously stated, with ODM, no data movement or conversion is needed. All the pre-and post-mining activities can be accomplished within the same environment. Your data is equally protected by the extensive security mechanisms of Oracle database as only users with the appropriate privileges can apply the mining models. The entire mining process is thus made less complex, time consuming and error-prone.

CHAPTER 3

3.0 PROJECT METHODOLOGY

3.1 Project Design

A review of the existing system and procedures at Blessed Superstores Limited is presented in this section, with an emphasis on the various activities carried out in the organization, in order to discover the problems in the system.

.

3.1.1 Review of the existing system

Blessed Superstores Limited is a business organization that deals on household goods and electrical equipment. The organization was established in 2005, in Umuahia, Abia State. It purchases and retails goods on cash basis. Transaction processing is purely manual. There are three basic departments in the organization, namely:

1. Stores
2. Accounts
3. Personnel

The Stores department is made up of three units as follows:

1. Sales Unit
2. Purchases Unit
3. Inventory control Unit.

The overall organizational structure is shown in figure 3.1

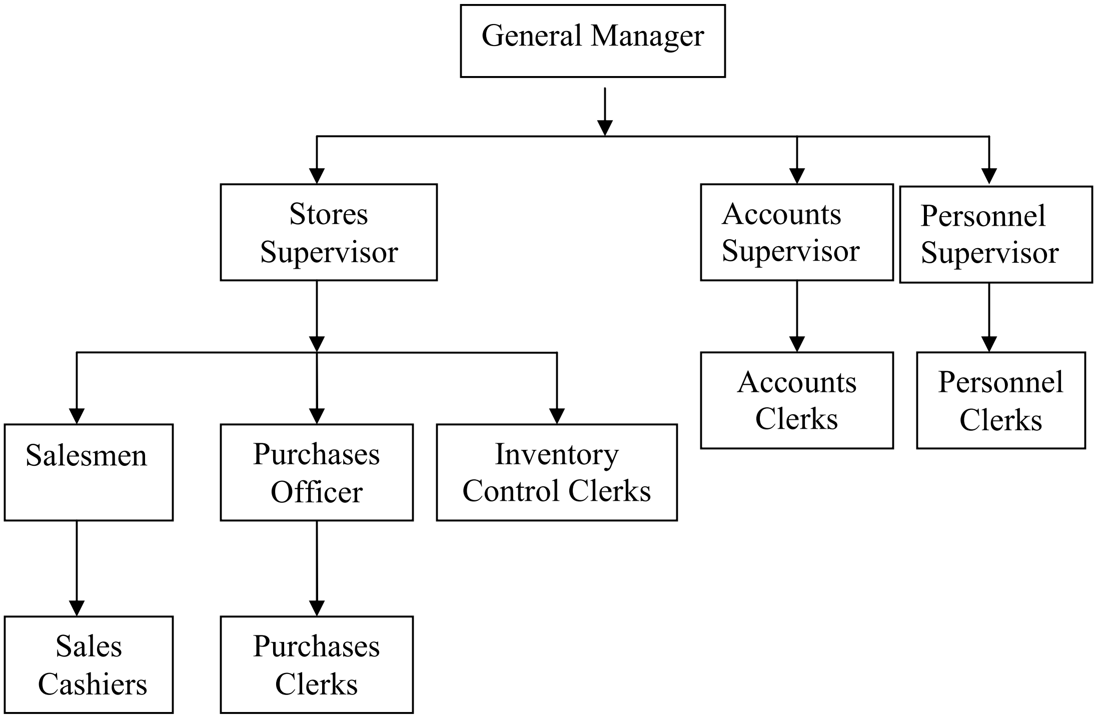


Figure 3.1: Organizational Structure of Blessed Superstores Limited

3.1.2 Functions Performed By Individual Departments

The functions performed by each department are as summarized below.

Stores Department

Sales

This unit is responsible for:

- Providing sales services to customers.
- Recording of sales data in the daily transaction sales book.
- Processing of sales data recorded in the transaction sales book.
- Recording of sales summary in the sales summary book.
- Generating sales reports.

Purchases

The following functions are performed in this unit:

- Purchasing of product items required for replenishment of stock ,and other items that might be needed in the organization.
- Recording of purchases data in the transaction purchases book.
- Processing of purchases data contained in the transaction purchases book.
- Generating purchases reports.

Inventory Control

The inventory control unit is responsible for making sure that proper inventory level is maintained on bin cards. It performs this function by;

- Continuously reviewing inventory levels. When the level reaches the reorder point, purchases unit is alerted, and a new order, whose size equals the economic order quantity, is placed.
- Carrying out periodic physical stock taking and adjusting any discrepancies on inventory level.
- Generating inventory reports.

Accounts Department

The accounts department keeps the financial records of the organization as follows:

- Records all cash expenditures in the expenditure cash book.
- Records all cash receipts in the receipt cash book.
- Maintains a cash accounts book.
- Generates reports.

Personnel Department

The personnel department handles all personnel matters relating to the organization as follows:

- Maintains various records about the employees. The records maintained include Employee personal records, Academic history records, Annual leave and Promotion records.
- Generates personnel reports.

3.1.3 Systems Inputs and Outputs

System Inputs

The inputs to the stores department are the cash sales invoice and purchases cash invoice which contain sales and purchases transactions respectively. Another input document is the Stock Adjustment Form, which contains data on either stock shortages or surplus. It is used to adjust inventory records on Bin Cards. Samples of the input documents are shown in Appendix A.

Systems Outputs

Output documents generated from the stores departments include the following:

Sales

Daily Sales Report Form: This is an output document generated by the sales unit. All the items sold in a day are contained in this report. The quantity of each item sold is extracted and used to update inventory records on Bin Cards.

Periodic Sales Report: This report is produced periodically, say weekly or monthly, and it contains periodic sales. The report has the same format as that of Daily sales.

Purchases

Daily Purchases Report Form: This is an output document generated by the purchases unit. The quantity of each item purchased is extracted and used to update inventory records on Bin cards.

Inventory Control

General Stock Properties Report: This report presents the general stock properties of the items held in stock.

Items below Reorder Level: This report contains items whose levels have fallen below predetermined reorder level.

Stock Prices: This report contains the selling prices of each item held in stock.

Samples of the output documents are shown in Appendix B.

3.1.4 Files and Records

The organization maintains a number of files and records on Bin cards, big notebooks, and forms filed and enclosed in office flat files. The files and records used in the stores department are discussed in this section.

Stores Department

Sales

Transaction Sales File: This file is used to record sales as they are made. The source document is the cash sales invoice. The file is maintained by a sales cashier in a big notebook. Each record in the file contains the following facts: Date, Part number, Item name, Type, Quantity sold, Unit price and Amount.

Daily Sales Summary File: This file is used to record summary of all the products sold in a day. The source data document is the transaction sales file, and it is also maintained by a sales cashier in a big notebook. Facts recorded in this file include: Date, Part number, Item name, Type, Quantity sold, Unit price and Amount.

Purchases

Transaction Purchases File: This file is used to record purchases data each time purchases are made. The source data document is the purchases invoice. A purchases clerk maintains the file in a big notebook. Each record in the file contains the following information: Date, Part number, Item name, Type, Quantity Purchased, Unit Cost and Amount.

Purchases Summary File: This file is used to record summary of purchases made in a day. The source data is the transaction purchases file. It is maintained by a purchases clerk in a big notebook, and each record in the file contains the following facts; Date, Part number, Item name, Type, Quantity Purchased, Unit Cost and Amount.

Inventory Control

Master Stock File: This file is maintained on bin cards by an inventory control clerk. Each product of trade has its own bin card. The quantity of any product item sold is extracted from daily sales reports and then subtracted from quantity on–hand, recorded on bin cards, to obtain the new stock on–hand.

Similarly, quantity of any item purchased is extracted from the daily purchase reports and added to the quantity on – hand, recorded on bin cards, to obtain the new stock on – hand. Another source of data to the bin card is the stock adjustment form. The data contained in this form is used to adjust the inventory records on a bin card. Each record in the file contains the following information: Part number, Item name, Type, Quantity on – hand, Unit Selling Price, Reorder level, and Economic order quantity.

3.1.5 Information Flow in Blessed Superstores Limited

From the flow diagram in figure 3.2, the double square is used to illustrate inputs to the system. The vertical rectangles show procedural steps, while the small flat rectangular boxes represent data flow which must be made available to that particular procedure.

3.1.6 Processing of the Existing System

The details of the manual processing system in the stores department at Blessed Superstores Limited are presented below.

Stores Department

Sales Unit

The sales cashier extracts sales data from the cash sales invoice and records it in a big notebook. At the end of the day, like products are sorted out, and calculations made to determine the total quantity of each product item sold, The results is then entered in the daily sales summary book. Daily and periodic sales reports are produced from this daily sales summary notebook.

Purchases Unit

Similar to that of the sales unit, the purchases clerk extracts purchases data from purchases invoices and records them in a transaction notebook. After all the entries have been made, like products are sorted out and calculations made to determine the total quantity of each product item purchased. The result is entered in the daily purchases summary notebook. Daily and periodic reports are produced from this notebook.

Inventory Control Unit

The inventory control unit records all the items of trade on a bin card. The processing in the inventory control unit is mainly updating of the inventory records. The daily sales and purchases reports received from sales and purchases units respectively are the main input documents. From the sales reports, the inventory control clerk extracts sales quantity for a particular product and subtracts it from quantity on-hand to obtain the new stock balance. Similarly, from the purchase report, the inventory control clerk extracts purchases quantity for a particular product and adds it to the quantity on-hand, to obtain the new stock balance. If during the stock update, the inventory control clerk notices stock shortages, he alerts the stores supervisor, who will then inform the purchase unit of the shortages for stock replenishment. Periodically, physical stock is carried out, any discrepancies are noted, and stock records are adjusted accordingly.

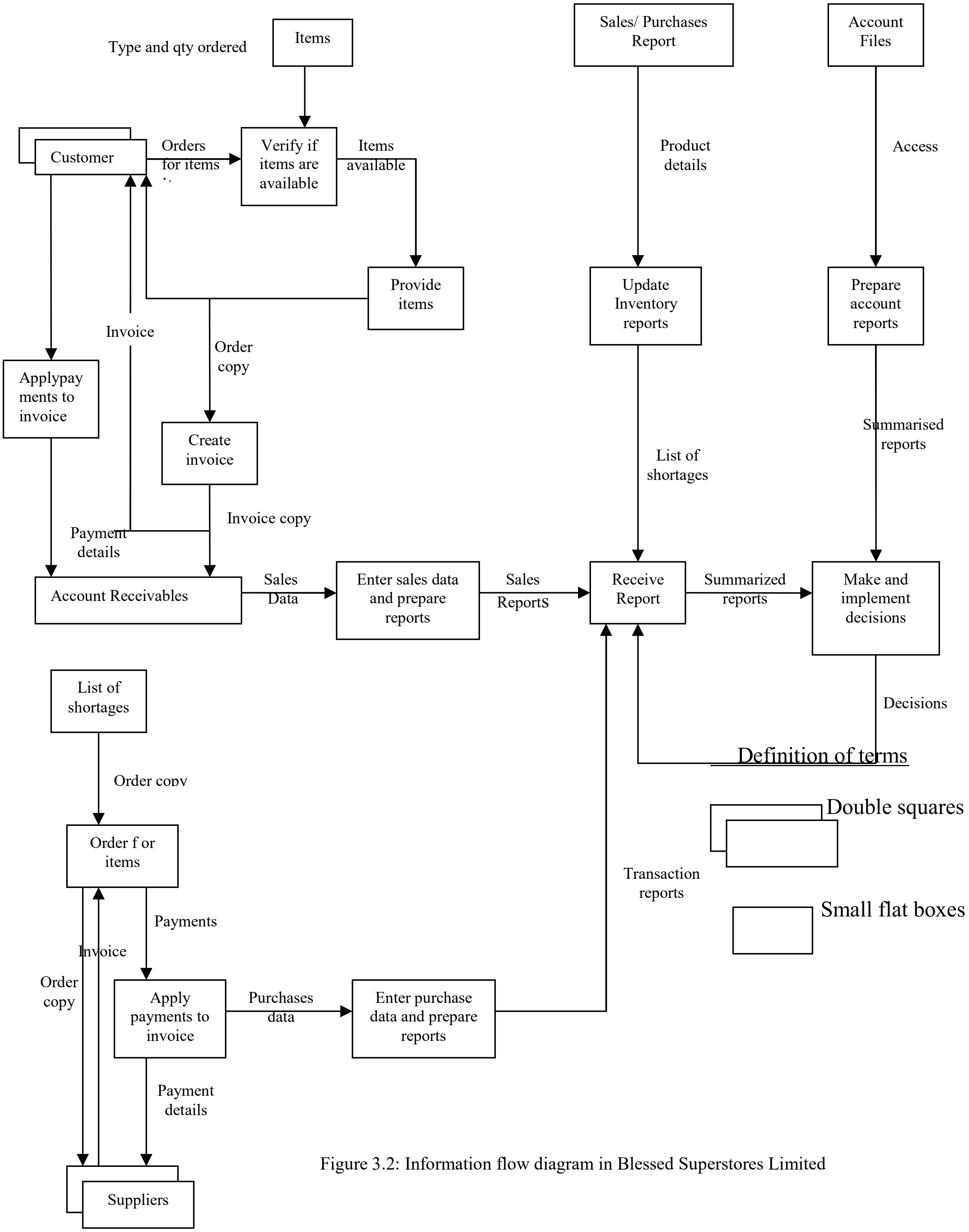


Figure 3.2: Information flow diagram in Blessed Superstores Limited

3.1.7 Problems of the Existing System

Interviews with the General Manager and the analysis of the existing system revealed the following problems and inadequacies:

1. Processing of the transaction data is tedious and time consuming. At times some errors are introduced into the system.

2. Reports are generated late.

3. Some of the reports are inaccurate because of errors introduced during processing. Any decisions that are taken based on these reports will also be wrong.

4. As a result of the delay in report generation, a number of management decisions are based on guesswork; a situation which has greatly affected the organization.

5. Maintenance of stock records on bins is tedious and time consuming too. As a result, proper inventory is not kept, resulting occasionally in stock outs.

6. The present system cannot answer on-the-spot ad-hoc queries.

7. The system can no longer cope with the peak period processing.

3.1.8 User Requirements

Having identified the shortcomings inherent in the present (manual) system, the user requirements, which were determined through interviews with both the management and staff of the various departments, are as listed below:

- **Timeliness of Information:** The new system must satisfy time requirements. Desired information must be produced at the right time.
- **Processing:** The new system must facilitate speedy processing of transaction data.
- **Accuracy:** Reports generated from the new system must be error free.

- **Stock Maintenance:** The new system must always maintain a reasonable stock at all times.
- **Queries:** The new system should be able to provide answers to on-the-spot ad-hoc queries.
- **Peak Period Processing:** The hardware to be chosen should be able to cope with peak period processing.

3.1.9 Samples of the New System Designs

The layouts of the computer output designs are shown in Appendix C, while the formats of the computer input designs are presented in appendix D.

3.2 Project Development

The following development steps are to be used for this project:

3.2.1 Point-of-Sale (POS) System Development

The POS System will be used to collect data from each item that is purchased at the superstore. The system collects data on the item brand name, category, size, time and date of the purchase, and at what price each of the items was purchased. All this data will be stored in a database, which would be data mined to uncover trends, relationships and patterns that can help the superstore manage its inventory more accurately, and generally optimize its business process.

The proposed software for the POS system shall be written, compiled and packaged to run on Windows operating system, to keep it in tune with the latest computer software technologies.

It shall be developed in Visual basic 6, which is a very popular, and powerful windows-based program.

3.2.2 The Database Management System

There are several methods of creating a database for use with Visual Basic [25]. The Visual Data Manager shall however be used for this work. The Visual Data Manager application that comes with Visual Basic provides one with an interactive way of creating and modifying databases. The application can be run by selecting Visual Data Manager Item from Visual Basic's Add-Ins menu. The Visual Data Manager can work with Access (Jet), dBase, FoxPro, Paradox and ODBC databases as well as text files.

Considering the problems of the existing system, and the user requirements specified in section 3.1.7 and 3.1.8 respectively, the relational database model shall be adopted.

Stores Database

The stores database is made up of tables that are normalized to separate items, which are independent of one another. However, the tables are related through the part number field. Normalization is used to avoid repetitions, while allowing for easy retrieval of information. The details of the stores database tables are discussed as follows:

Stock Table

This table contains records on stock traded. The items of information recorded in the table should be part number, item name, item type, quantity on–hand, unit selling price, reorder level, and economic order quantity. As transactions are made, the quantity on–hand should automatically be updated to reflect the correct quantity of items held in stock.

Sales Table

This table is a transaction table, used to record all product items sold. The items of information recorded in the table include; date, part number, quantity sold, unit selling price, and amount.

Purchases Table

This table is a transaction table used to record all product items purchased. It has the following data fields; date, part number, quantity purchased, unit cost, and amount.

Password Database Table

The password database table records the particulars of the bonafide users of the system. It has the following data fields; user identification, user name, access level, and department.

Login Database Table

The login database table contains the records of people that make use of the system at a particular point in time. It has the following data fields; password, user name, department, logdate, logtime, and logout.

Database Table Structure

The following information are given for each of the database tables listed above

- Table name
- Table type
- Table structure
- Indexes defined

Stores Database Tables

Stock Table

Table name: STOCK.DB

Table type: Paradox

Table 3.1 Structure of Stock table

S/N	Field Name	Field type	Size	Key
1	PartNo	C	5	*
2	ItemName	A	25	
3	ItemType	A	12	
4	Qnty-On-hand	S	5	
5	UnitSellPrice	N	9	
6	ReordLevel	S	5	
7	EconOrderQnty	S	6	

Indexes

The table is indexed on PartNo field as indicated by the key column.

Sales Table

Table name: SALES.DBF

Table type: dBase IV

Table 3.2: Structure of Sales table

S/N	Field name	Field type	Size	Key
1	SaleDate	D	10	*
2	PartNo	C	5	0
3	QntySold	N	7	0
4	UnitPrice	N	10	2
5	Amt	N	12	2

Indexes

SalesNdx -indexed on SalesDate field.

Purchases Table

Table name: PURCHASES.DBF

Table type: dBase IV

Table 3.3: Structure of Purchases table

S/N	Field name	Field type	Size	Key
1	TraDate	D	10	*
2	PartNo	C	5	0
3	QntyPurch	N	6	0
4	UnitCost	N	9	2
5	Amount	N	10	2

Indexes

DateNoNdx - indexed on TraDate field

Password Table

Table name: PASSWORD.DB

Table type: Paradox

Table 3.4: Structure of Password table

S/N	Field name	Field type	Size	Key
1	UserId	A	10	*
2	UserName	A	25	
3	AccessLevel	S	1	
4	Department	A	25	

Indexes

Indexed on UserId

Login Table

Table name: APPLPO.DBF

Table type: dBase IV

Table 3.5: Structure of Login table

S/N	Field name	Field type	Size	Key
1	Password	C	10	
2	UserName	C	25	
3	Department	C	25	
4	LogDate	D	10	*
5	LogTime	C	10	
6	Logout	C	10	

Indexes

Indexed on LogDate

3.2.3 System Processing and Controls

The system flowchart shown in figure 3.3 depicts the stores processing operations within the proposed system. A keyboard is used as the input device for data entry into the computer. During processing, various data are displayed and viewed on – line through the screen. The database files are maintained on a magnetic disk, with backups on tapes. Different transaction runs produce different reports which can be viewed on screen, or printed out. Inquiries can be made from the database, and appropriate data displayed, and viewed on – line through the screen. All the computer processes, with the related documents are equally represented.

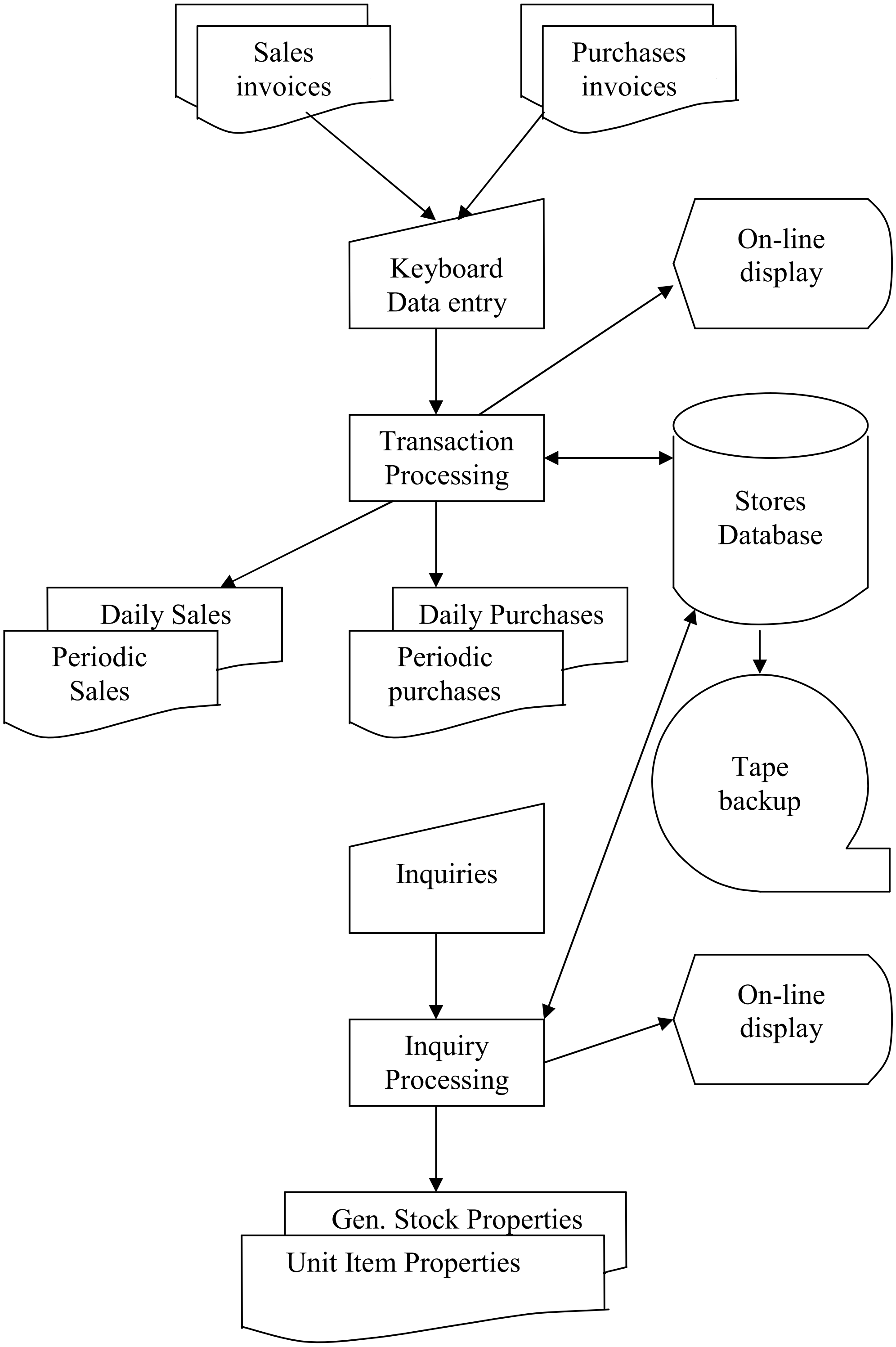


Figure 3.3: Stores Transaction Processing System flowchart

System Controls

System controls are used to ensure that only valid data are accepted and processed completely and accurately by the computer system. Controls usually take the form of verification (by people) and validation (by software) procedures.

Data Verification

Before data can be input into the computer system, it has to be verified for correctness. If the source data is not correct, or does not conform to the specified format, appropriate corrections would have to be made. To this end, all transaction tables in this design will use date field, and certain searches in the database will also be based on the values contained in this field. It is therefore important that the date values contained in this field be properly verified and entered in the correct format . In this design, the format for all the date fields would be **dd /mm /yyyy,** where dd are two digits, representing the day of the month, [01 – 31], mm are another two digits, representing the month numbers, [01 – 12]. yyyy are four digits, representing the year. / is a forward slash, indicating a date separator.

The above format means that each date must contain 10 characters; the date separator character must be the forward slash character; days and months, with numbers less than 10 must be represented with the zero prefixes. For example; 01, 02, 03, 04 - - - - 09.

Data Validation

Data validation is carried out, using computer programs to check if the input data is correct. The main types of checks are the range and value checks. Exception – handling routines should be coded to help the user correct any intercepted input error. Furthermore, Windows input functions, utilizing combo and list boxes should be used to control both range and value input errors.

3.2.4 System Security

To prevent unauthorized access to database files, adequate system security measures have to be provided. System security means, ensuring that data is inaccessible to unauthorized personnel. It also incorporates all database securities, relating to dumping of database data into a magnetic tape to safeguard against accidental erasure or corruption of the data. The following security measures shall be applied in the new system.

Use of Password

A password should be provided to the bona fide users of the applications. The password, when input to the system, is compared with that stored in the operational software. Access is barred if the password entered is incorrect.

Use of Access Level

Access to some parts of the database shall be restricted to specified users. That is, users can only access software functions for which they are responsible.

Back–up and Recovery

As provided in the design, the database should be dumped periodically into a magnetic tape. Whenever it becomes necessary, the database can be easily reconstructed by copying the data stored in the security tape, back to the database.

Database Clean - up

Due to the limited storage capacity of magnetic disks, a utility program would be incorporated in the program to perform the database clean up. A copy of the database would be made before performing the database clean up.

3.2.5 Description of the program structure

The general program structure is as shown in Fig. 3.4

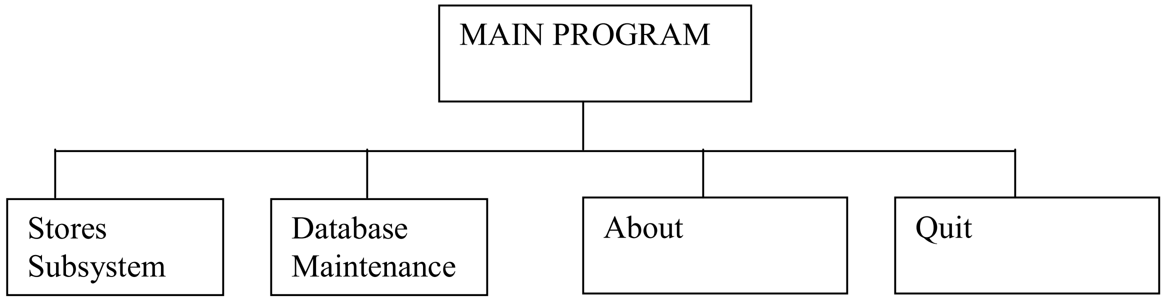


Figure 3.4: General program structure.

Figure 3.4 shows the highest level of the program structure. The Stores subsystem handles all stores functions (sales, purchases and inventory control), while the Maintenance option handles the database maintenance functions which include, database back-up/restore, database clean-up and password administration. It also incorporates a program function that enables one view user's login. The About option displays information about Blessed Superstores, and the Software, while the Quit option enables you to quit the program. The program listing is presented in Appendix E.

The Stores Subsystem

Fig 3.5 is a schematic representation of the stores subsystem program structure

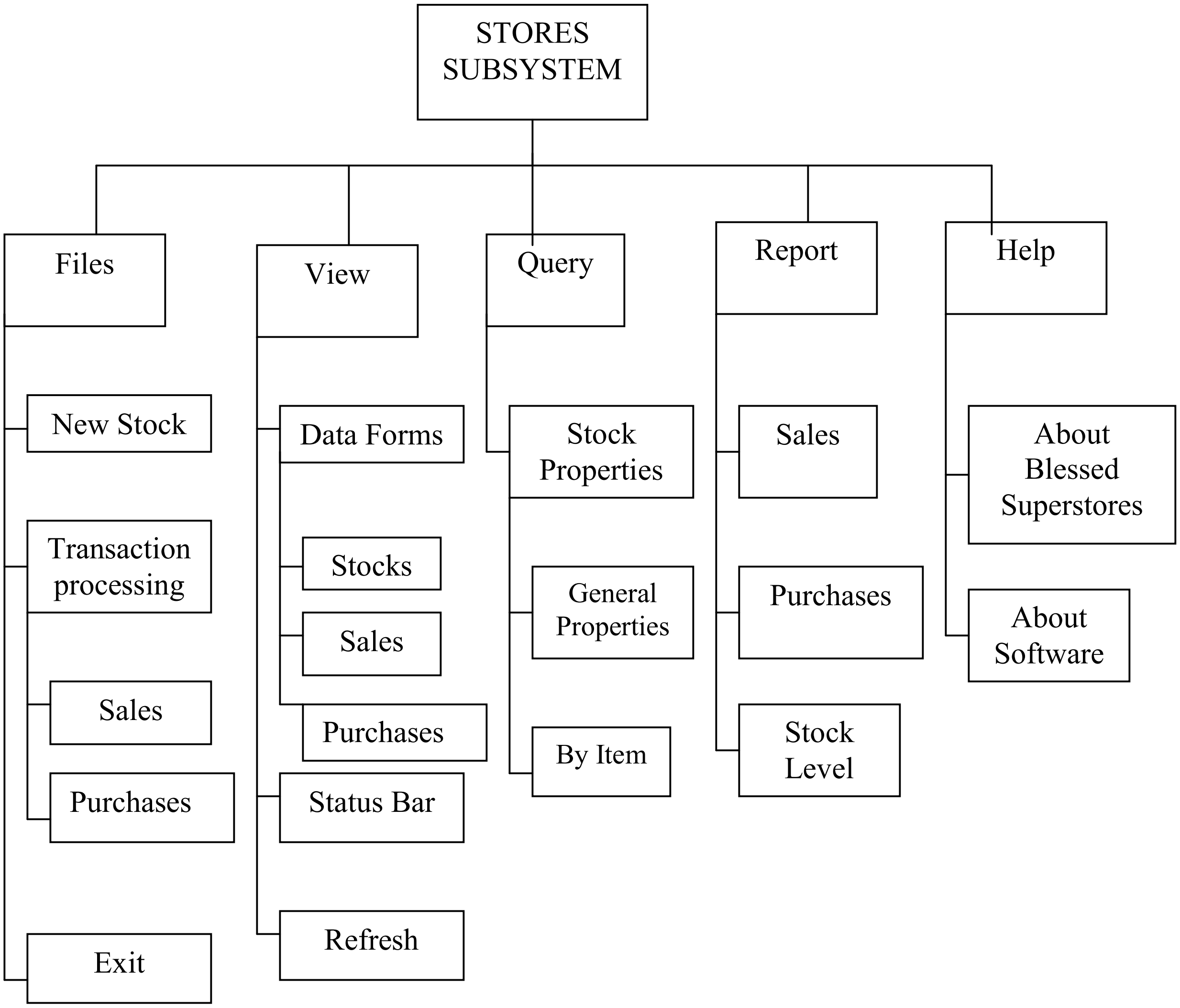


Figure 3.5: Schematic representation of Stores subsystem program structure

The first level of the stores subsystem program structure is the subsystem program menu which contains File, View, Query, Report and Help. Each menu can further be expanded to other submenus as shown in the figure.

Database Maintenance

The database subsystem handles the database maintenance and the program structure is as shown in figure 3.6.

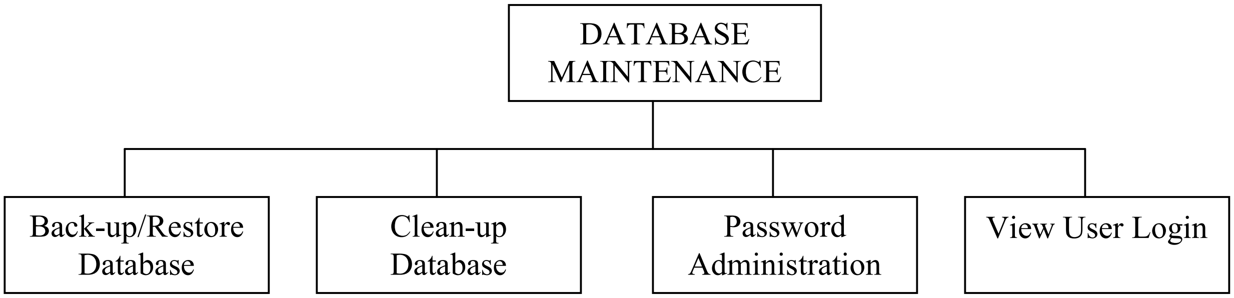


Figure 3.6: Database maintenance subsystem program structure.

3.2.6 The Password Access Levels

Three access levels of password are to be implemented.

Access level 1

Individuals with this access level can access all parts of the software and can equally update any file. It is normally the person in charge of the computing resources or the Database Administrator.

Access level 2

Individuals with this access level can only access the functions in the subprogram designed for his unit/ department. Three units are recognized as follows:

STR-Stores

SLE-Sales

CPS- Computer Services-Maintenance Subsystem

Thus an individual with access level of 2 in Stores unit can only access all the functions in the stores subprogram. There is however an exception to this rule in the CPS- Computer Services Unit. With access level of 2, an individual can access all functions in the Maintenance subsystem, with exception of the Password Administration option. This option can only be accessed by individuals with the access level of 1.

Access Level 3

Individuals with access level of 3 are restricted to their units. However, they are not allowed to update/ add to or delete the contents of any database tables.

User passwords are encoded before being stored in the password table. The following algorithm is used. For each character of the user password, the encoding function uses the next two characters in the collating sequence. For example, a user with password “Jane” is translated to “Lcpg’. Again a password “John” is translated to “Lqjp”. The following program listing shows the password encoding function.

```
Private Function Code Password$(X$)
    Pass$ =””
    For 1$ = 1 To Len(X$)
        Ch$ =Mid$(X$,1%,1)
        Pass$ = Pass$ + Chr$ (Asc(Ch$)+2)
    Next
    Code Password$ = Pass$
End function
```

CHAPTER 4

4.0 PROJECT IMPLEMENTATION

4.1 Point-of-Sale System Implementation

The Visual Basic source codes for the implementation of the POS are presented in Appendix E.

4.2 Program Installation

The following steps are to be followed in installing the program:

1. Click the Start button at the left corner of the desktop taskbar and select Program from the menu that appears. Select and click Windows Explorer from the program menu.
2. Select and click a drive where a folder is to be created.
3. Select and click File to open the file menu. Select and click New for the New item menu to appear. From the New item, select and click Folder and then type the name of the folder. The name of the folder is BLESSED.
4. Insert the CD ROM containing the software in drive E and click the drive.
5. In the list of files displayed, position your mouse pointer to the first file menu.
6. Click the Edit menu and then select and click Select all. All the program files shall be highlighted, indicating their selection.
7. Click the Edit menu again, select and click Copy.
8. Open the folder where the program files are to be copied by clicking BLESSED.
9. Once more, click the edit menu, then select and click Paste for the program files to be copied to BLESSED.

4.2.1 Running the program

The following steps are taken to run the application software.

1. At the desktop, double click BLESSED. This starts the application.

2. The Password Window appears for you to enter your password.

3. Enter your password and click OK button. If a valid password is entered, the main program interface appears. Three options are available for selection. The options are Stores subsystem, Maintenance and Quit.

Using the Stores Subsystem

The Stores Subsystem implements services offered by the stores department. Selecting Stores Subsystem from the application's main program interface will present the Stores Subsystem main menu. The menu items presented are File, View, Query, Report and Help. Each menu contains submenus that can be pulled down.

File menu

The File menu contains all tasks in which the contents of any of the database tables may be changed in any way. These, among other things include entering of new stock to the database and processing of sales/purchases transactions.

To execute any task from this menu

1. Click the File menu to pull down the menu

2. Click the item you want to execute and follow the instructions specified.

3. Selecting transaction processing will open the underlying submenu. Select option of your choice and click. Follow the instructions specified to perform the task.

At the end of each task, control returns to the subsystem menu. Selecting and clicking the Exit option closes the application subsystem and returns to the mains application interface.

View Menu

This menu contains options that will enable you view the contents of the database tables. To execute any task,

1. Click the View menu to open the menu.

2. Click the item you want to execute and follow the instructions specified.

3. Selecting and clicking Data forms will open an underlying submenu.

4. Click any item of your choice to execute.

Query Menu

The Query menu contains tasks that can be used to query the database. Some of these tasks are contained in the submenu. To execute any query task,

1. Click the Query menu to open the menu.

2. Click any item you want to query and then follow the instructions specified.

Report Menu

The Report menu contains options that will be used to generate reports from the various activities handled by the application subsystem. The tasks handled are contained in the submenu associated with the Report menu. To execute any task,

1. Click the Report menu to open the menu.

2. Click any Report item you want to generate and follow any instruction specified.

Help Menu

The Help menu will give you an opportunity to get information about Blessed Superstores and the software. To perform this task,

1. Click the Help menu

2. Select and click any item you require for its information.

Quitting the Application Software

To quit the application software, and return to the desktop,

Click the Quit button.

4.3 System Testing

The following tests were conducted in the course of this work.

Data Range Testing

This test is conducted to ensure that the expected data range is maintained. For example, deliberate attempts were made to enter out-of-range date values into the database and this error was detected.

Duplication of values in Key field

Duplicate values in key fields often lead to corruption of the database. The master database files were implemented and tested to ascertain that no duplicate values in key fields exist. An error trapping program incorporated in the application software is meant to take care of this.

Database Access

Here, tests were conducted to verify that unauthorized users should not have access to the database. The password administration subprogram performs this function. Deliberate attempts were made to have access to the database through the use of wrong passwords and access was not granted. Only the authorized users with authentic passwords could have access to the database.

CHAPTER 5

5.0 CONCLUSION

Data mining has been described as one of the greatest technologies to be introduced in the retail industry in recent times. It greatly assists organizations uncover patterns hidden in their data that can be used to predict the behavior of customers, and monitor the condition of products. Today, most enterprises are actively collecting and storing large databases. Many of them have recognized the potential value of these data as an information source for making business decisions.

However, despite its rapid growth, data mining is still an evolving field. Empirical research in data mining applications in the retail industry is limited. Studies have been focusing mainly on the e-commerce sector. For instance, data mining has been successfully explored in optimizing inventory levels for electronic commerce and to analyze product performance on online stores. Researches in the non on-line sector are very much limited. Clearly, there is much more to be accomplished with this very powerful research tool.

Again, there is still a growing gap between the more powerful storage and retrieval systems, and the user's ability to effectively analyze and act on the information they contain. Data mining tools need to be guided by users who understand the business, the data, and the general nature of the analytical methods involved. Retailers are usually concerned with the application of data mining results than in the way it works. They need simple –to-use tools that can efficiently solve their business problems. Existing software packages lack sufficient support for both directing the analysis process and presenting the analysis results in a user-friendly manner.

- **CONTRIBUTION**

The application of data mining for enhanced decision making has only been optimally utilized in the electronic-commerce sector and big retail firms because of its prohibitive cost. This work has however focused on its use by small retail firms to boost sales and significantly increase revenue. The comparative analysis of the different analytic methods available for decision making, presented in section 2.3.4, shows that data mining provides the competitive difference in retailing by delivering the predictive insight needed by retailers to stay ahead of the competition.

5.1 RECOMMENDATION

The adoption of a data mining system in Blessed Superstores Limited was necessitated as a result of the problems inherent in the existing system. Even though it is still at the introductory stage, it is believed that an effective implementation of the scheme will improve the overall performance of the organization's business activities.

It is therefore recommended that anyone wishing to expand the scope of this work should be allowed to do so. While there may be fundamental problems that remain to be solved, there have also been numerous success stories recorded, and the benefits of data mining technology have been convincingly demonstrated in the broad range of application domains. As electronic databases continue to grow, data mining will continue to make advances, producing predictive information and helping businesses leverage their data to facilitate more effective and efficient proactive decision-making.

REFERENCES

[1] Berry, M.J.A., and Linoff, G.S. (1997); "Data mining techniques for marketing, sales and customer support", John Wiley & Sons, Inc. New York. pp.20-35

[2] Thearling, K., (2007); "An Introduction to Data Mining: Discovering hidden values in your data warehouse". Excerpt: White Paper: Pp.1-6 http://www.thearling.com/text/dmwhite/dmwhite.htm

[3] Rabinovitch, L., (1999, December); "America's first department store mines customer data", Direct Marketing, volume 62, issue 8, pp 42-44

[4] Monk, E., and Wagner, B. (2006); "Concepts in Enterprise Resource Planning", Second Edition. Thomson Course Technology, Boston. ISBN: 0-619-21663-8

[5] Berson, A., Switch, S., and Thearling, K. (1999); "Building Data Mining Application for CRM", McGraw-Hill. Excerpt: White Paper: pp. 1-10 http://www.thearling.com

[6] Oliver, E.C., Champion, R.J., and French, C.S. (1991); "Data Processing and Information Technology", Eight Edition. DP Publications. pp.1-270

[7] Lucey, T. (1991); "Management Information System", Sixth Edition .ELBS, DP Publications. pp.1-274

[8] Lee, B.S. (1990); "Systems Analysis", Stanley Thornes Publishers, England. pp. 5-10

[9] Kuonen, D. (2004, October); "Data Mining and Statistics: What is the Connection?", TDAN.com. pp.1-6

[10] Decker, P. (1998, March); "Data Mining's hidden dangers: Banking Strategies", volume 74, issue 2, pp.6-14

[11] "Data Mining Concepts", (2008); SQL Server 2005 Books Online, pp.1-6

(Retrieved from http://www.w3schools.com/sql/sql server.asp)

[12] Anderson, J. and Kotsiopulos, A. (2002); “Enhanced Decision Making using Data Mining: Applications for Retailers”, Journal of Textile and Apparel, Technology and Management. Volume 2, issue 3, pp.1-13

[13] Parks, L. (2000,June); “Information Resources Inc.”, Drug Store News, volume 22, issue 9, pp.219-222

[14] Whiting, R. (2001, November); “Retailers seek success on how customers dress”, Information Week, volume 865, pp.65

[15] Rabinovitch, L. (1999, December); “America’s first department store mines customer data”, Direct Marketing, volume 62, issue 8, pp.50-54

[16] Data Intelligence Group, (1995, September); “An Overview of data mining at Dun and Bradstreet”, http://www3.shore.net (retrieved June 11, 2010).

[17] Thearling, K. (2002); “More than algorithms: Data mining in the real world” http://www.thearling.com (retrieved June 11, 2010).

[18] Anonymous (1996); “Do-it-yourself data mining”, Marketing Computers. pp.1-2

[19] Stedman, C. (1997, December); “Data Mining despite its dangers”, Computer World, volume 32, issue 1, pp.61-62

[20] Goebel, M., and Gruenwald, L.A. (1999); “A Survey of Knowledge Discovery and Data Mining Tools”, Technical Report, SIGKDD Explorations. Volume 1, issue 1, pp.1-50

[21] Elder, J.F., and Abbott, D.W. (1998, August); “A Comparison of Leading Data Mining Tools”, Fourth International Conference on Knowledge Discovery & Data Mining. New York. Pp.1-10

22] Robert, A.N. (2002); “Data Mining Review” - White Paper http://www.dmreview.com/channels/data mining.html

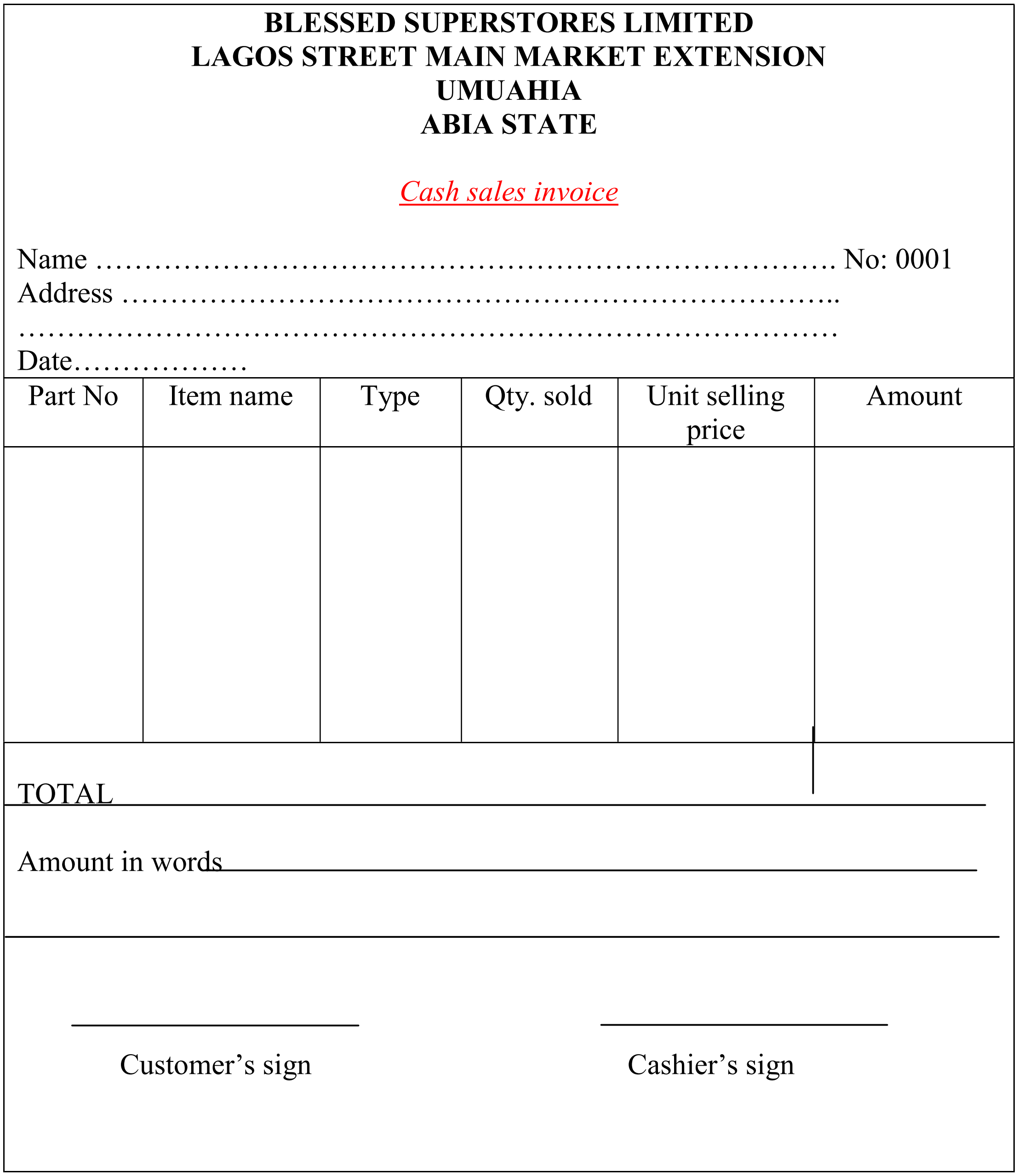
[23] “Data Mining”, Wikipedia, the free encyclopedia, pp.1-5

(Retrieved from http://en.wikipedia.org/wiki/Oracle_Data_Mining)

[24] “Oracle Data Mining Concepts 11g Release 1”, (2008); Oracle Online Data Mining Documentation Library, pp.83-107 (Retrieved from http://www.Oracle data mining.com)

[25] Siler, B., and Spotts, J. (1998); “Using Visual Basic 6”, Special Edition. Que. U.S.A. pp.55-661

APPENDIX A: SAMPLES OF THE MANUAL SYSYEM INPUTS



BLESSED SUPERSTORES LIMITED
LAGOS STREET MAIN MARKET EXTENSION
UMUAHIA
ABIA STATE

Cash sales invoice

Name …………………………………………………………………. No: 0001
Address …………………………………………………………………..
……………………………………………………………………………
Date………………

Part No	Item name	Type	Qty. sold	Unit selling price	Amount

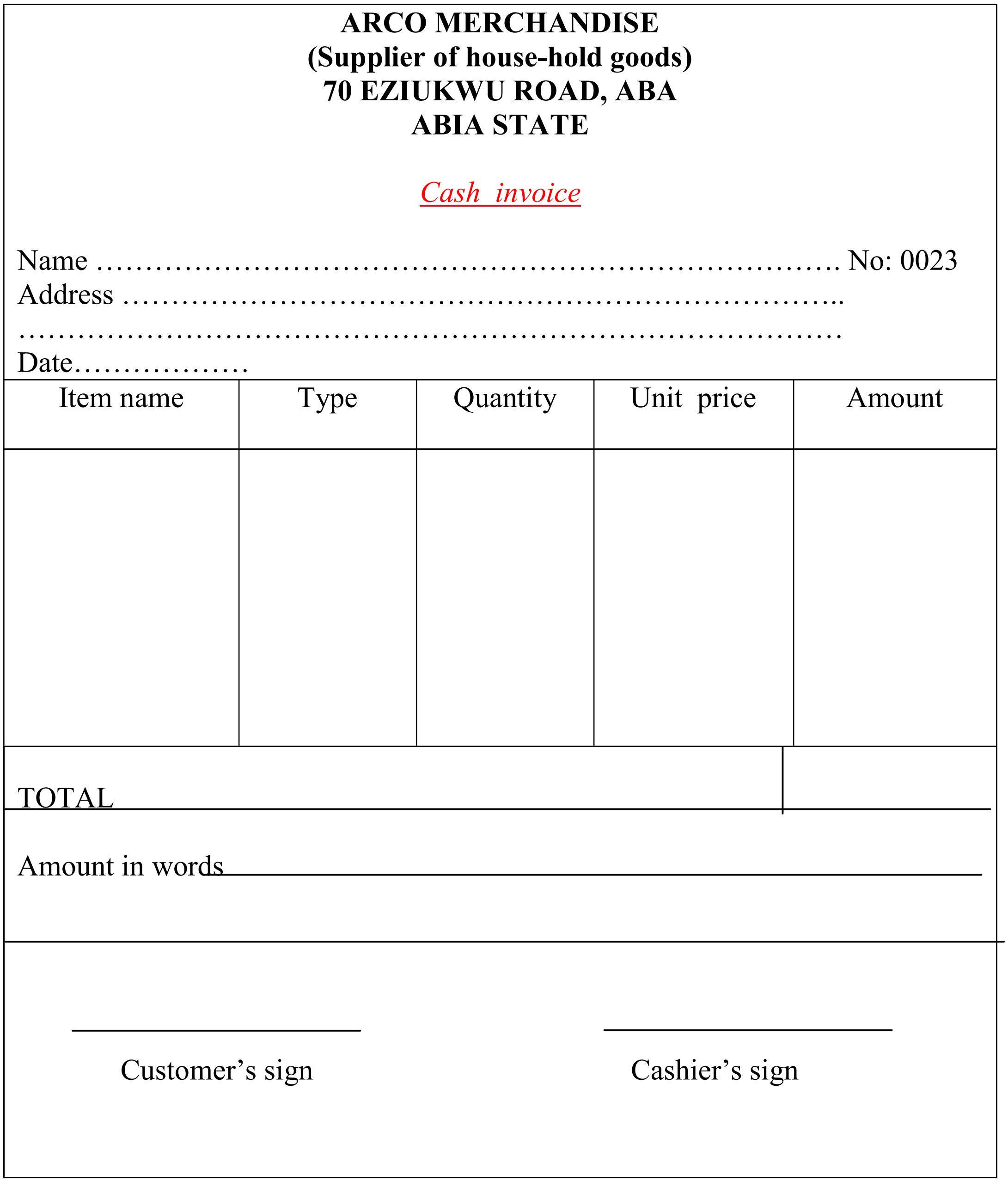
TOTAL

Amount in words

Customer's sign

Cashier's sign

Figure A.1: Cash sales invoice form format



ARCO MERCHANDISE
(Supplier of house-hold goods)
70 EZIUKWU ROAD, ABA
ABIA STATE

Cash invoice

Name ……………………………………………………………………. No: 0023
Address …………………………………………………………………..
……………………………………………………………………………
Date………………

Item name	Type	Quantity	Unit price	Amount

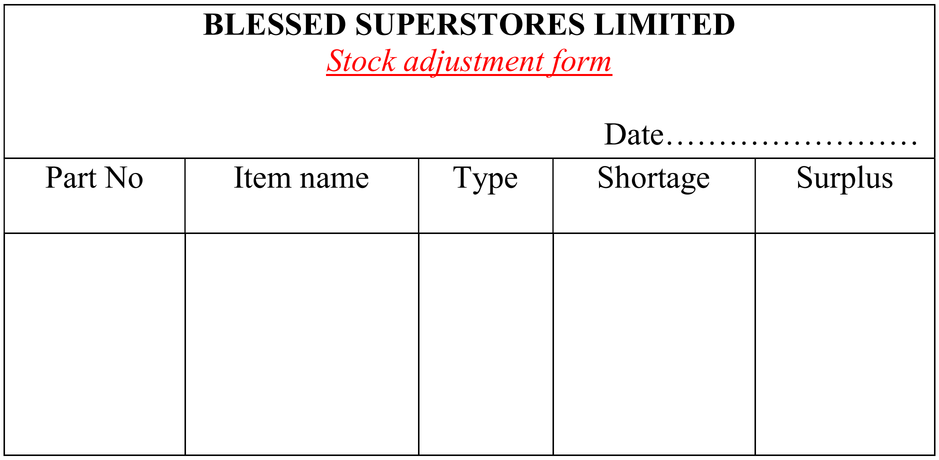
TOTAL

Amount in words

Customer's sign

Cashier's sign

Figure A.2: Purchases cash invoice form format



BLESSED SUPERSTORES LIMITED

Stock adjustment form

Date……………………

Part No	Item name	Type	Shortage	Surplus

Figure A.3: Stock Adjustment form format.

APPENDIX B: SAMPLES OF THE MANUAL SYSTEM OUTPUTS

BLESSED SUPERSTORES LIMITED

Daily Sales Report

Date……………………

Part No	Item name	Type	Qty sold	Unit price	Amount

Figure B.1: Daily Sales Report form format

BLESSED SUPERSTORES LIMITED

Daily Purchases report

Date……………………

Part No	Item name	Type	Qty purchased	Unit cost	Amount

Figure B.2: Daily Purchases Report form format

BLESSED SUPERSTORES LIMITED

General Stock Properties Report

Date……………………

Part No	Item name	Type	Qty on hand	Unit price	Reorder level	Econ. Order Qty.

Figure B.3: General Stock Properties_Report form format

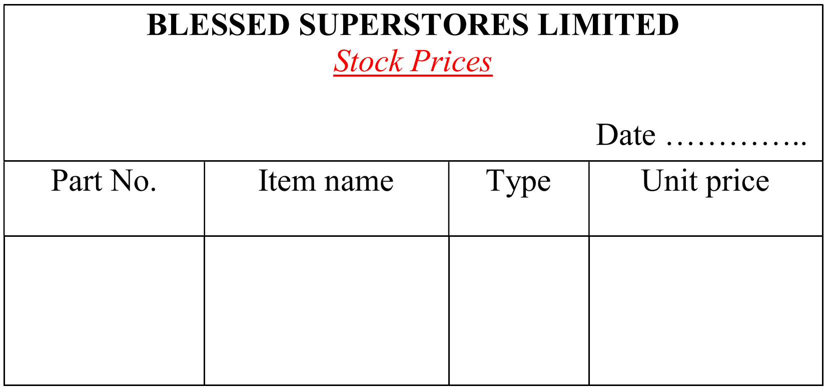
BLESSED SUPERSTORES LIMITED

Items Below Reorder Level

Date …………..

Part No.	Item name	Type	Qty on hand	Unit price	Reorder level	Economic Order qty.

Figure B.4: Items Below Reorder Level form format



BLESSED SUPERSTORES LIMITED

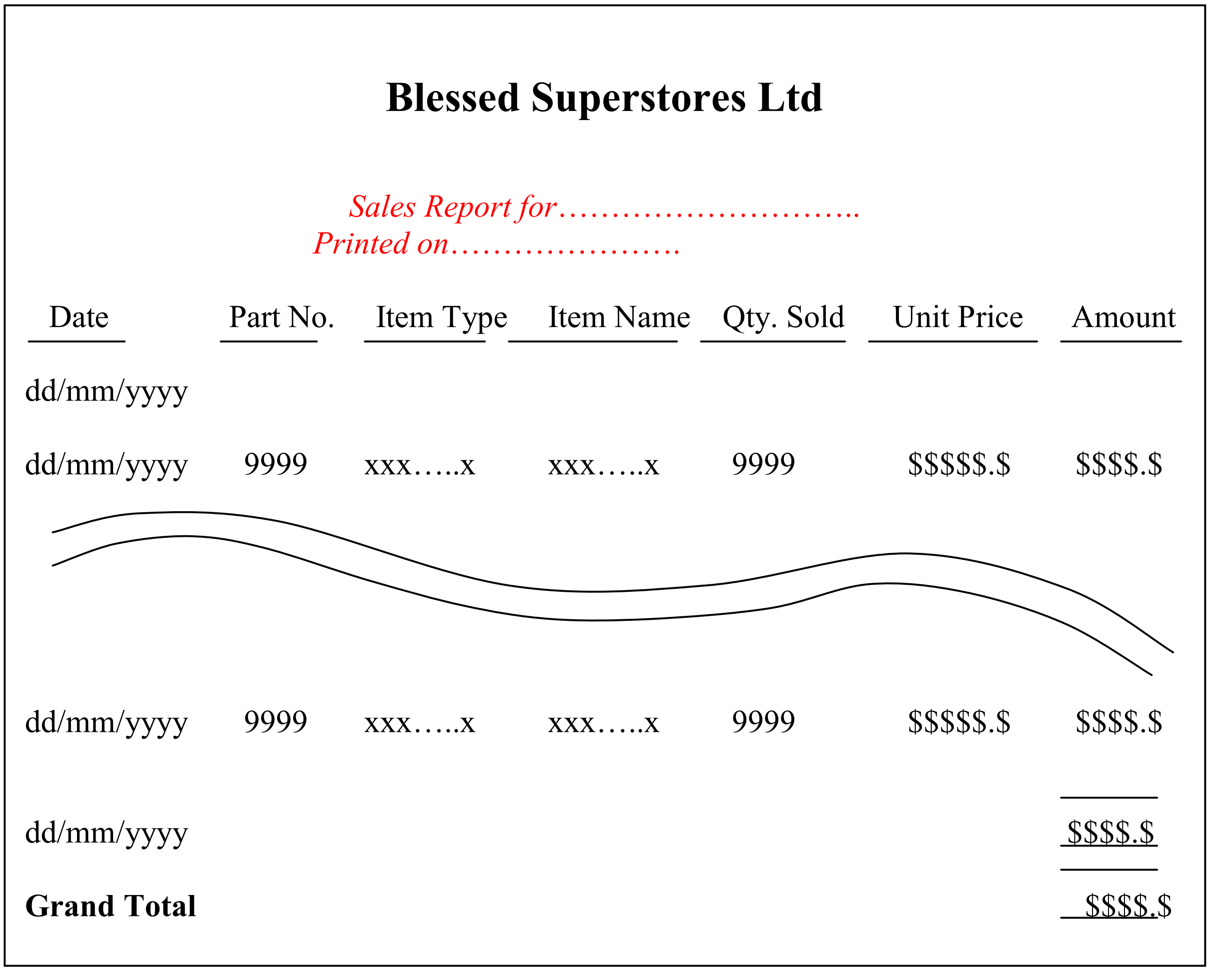
Stock Prices

Date …………..

Part No.	Item name	Type	Unit price

Figure B.5: Stock Prices Report form format

APPENDIX C: LAYOUTS OF THE COMPUTER OUTPUT DESIGNS

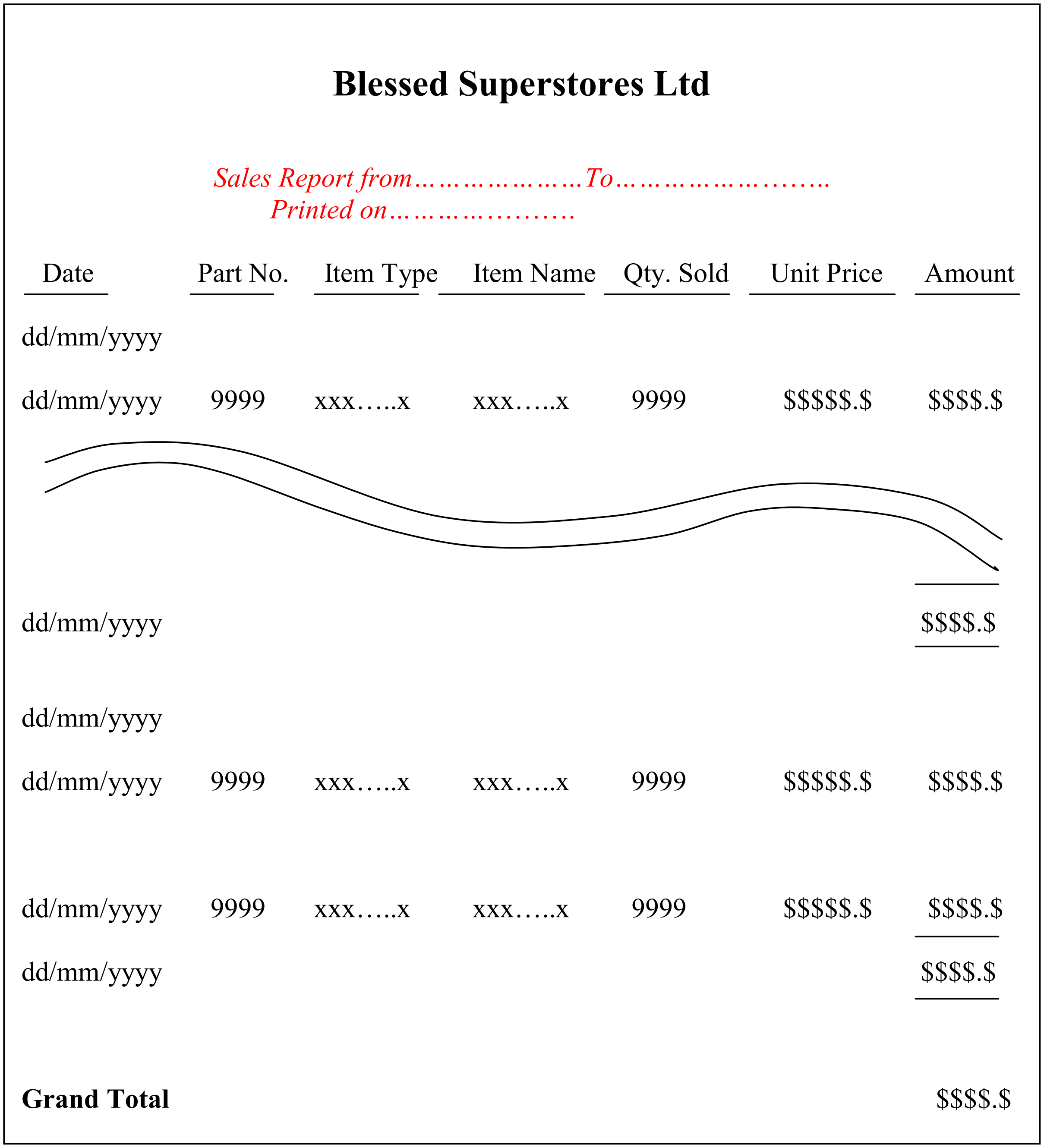


Blessed Superstores Ltd

Sales Report for………………………..
Printed on………………….

Date	Part No.	Item Type	Item Name	Qty. Sold	Unit Price	Amount
dd/mm/yyyy						
dd/mm/yyyy	9999	xxx…..x	xxx…..x	9999	$$$$$.$	$$$$.$
dd/mm/yyyy	9999	xxx…..x	xxx…..x	9999	$$$$$.$	$$$$.$
dd/mm/yyyy						$$$$.$
Grand Total						$$$$.$

Figure C.1: Layout of daily sales report

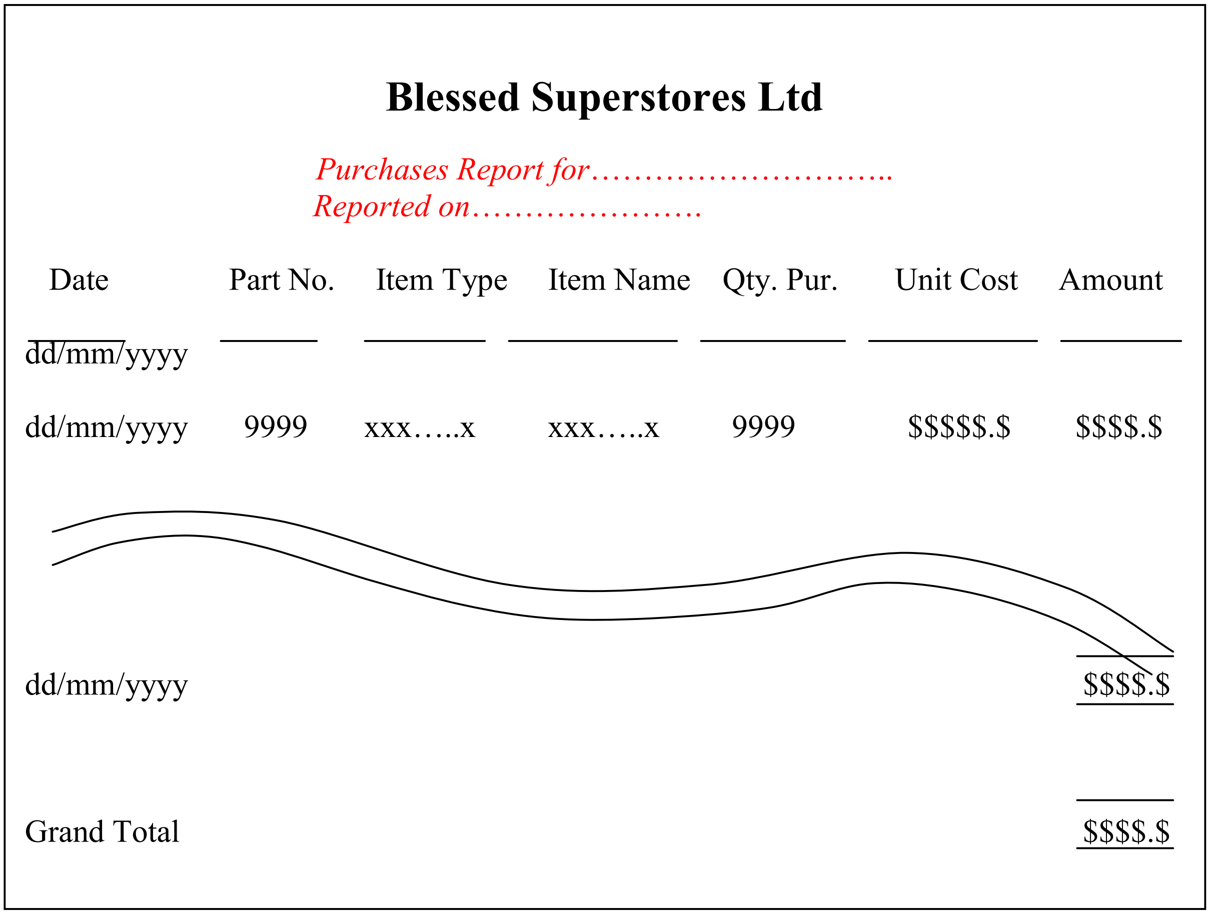


Blessed Superstores Ltd

Sales Report from…………………To………………………..
Printed on………………….

Date	Part No.	Item Type	Item Name	Qty. Sold	Unit Price	Amount
dd/mm/yyyy						
dd/mm/yyyy	9999	xxx…..x	xxx…..x	9999	$$$$$.$	$$$$.$
dd/mm/yyyy						$$$$.$
dd/mm/yyyy						
dd/mm/yyyy	9999	xxx…..x	xxx…..x	9999	$$$$$.$	$$$$.$
dd/mm/yyyy	9999	xxx…..x	xxx…..x	9999	$$$$$.$	$$$$.$
dd/mm/yyyy						$$$$.$
Grand Total						$$$$.$

Figure C.2: Layout of periodic sales report



Blessed Superstores Ltd

Purchases Report for………………………..

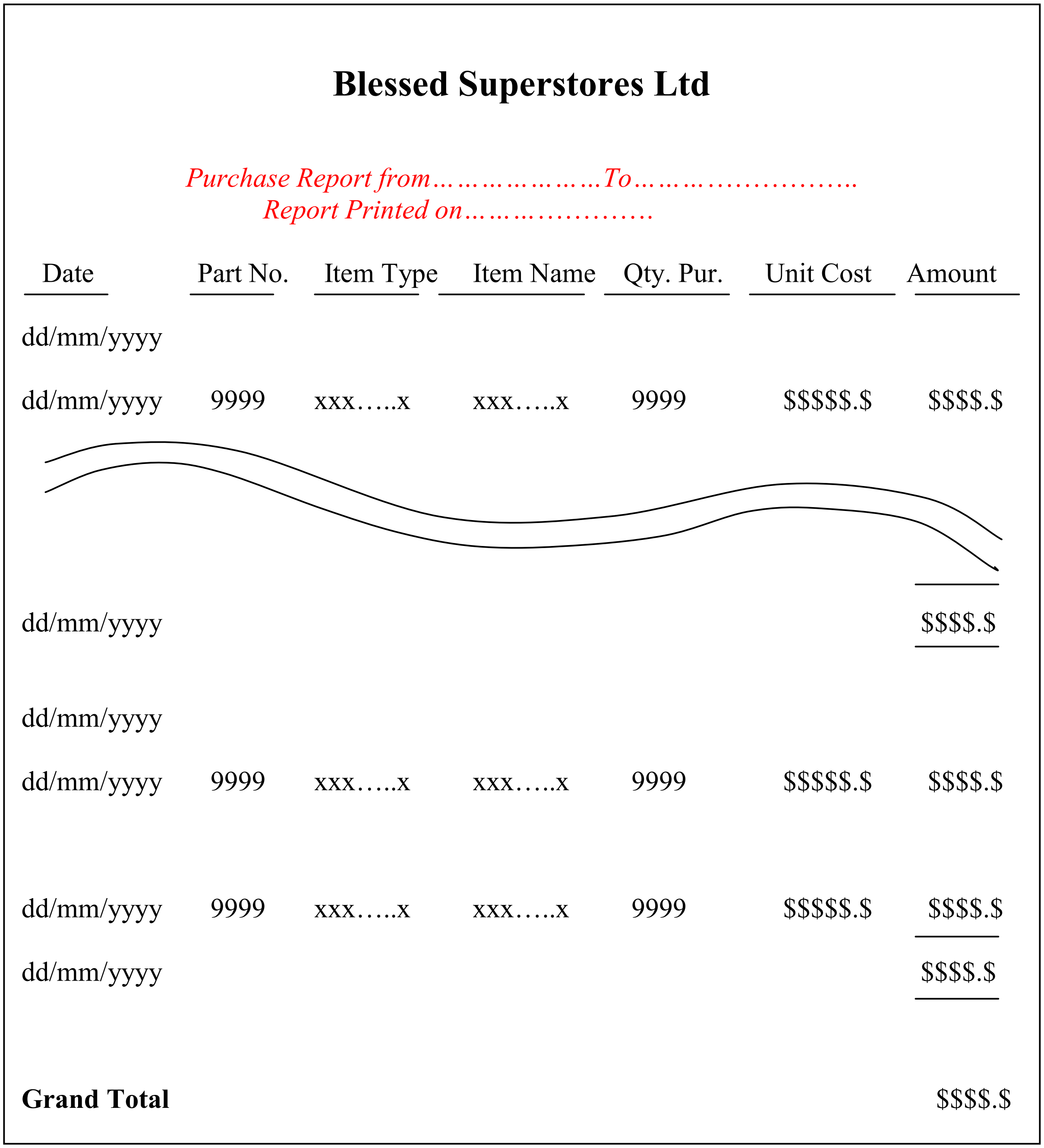
Reported on…………………

Date	Part No.	Item Type	Item Name	Qty. Pur.	Unit Cost	Amount
dd/mm/yyyy						
dd/mm/yyyy	9999	xxx…..x	xxx…..x	9999	$$$$$.$	$$$$.$
dd/mm/yyyy						$$$$.$
Grand Total						$$$$.$

Figure C.3: Layout of daily purchases report

STOCK LEVEL					
Part No	Item Name	Item Type	Qty on Hand	Reorder level	Econ. Order Qty
9999	xxxxxxxxxxx	xxxxxxxxxxx	999999	99999	999999

Figure C.4: Layout of stock level report



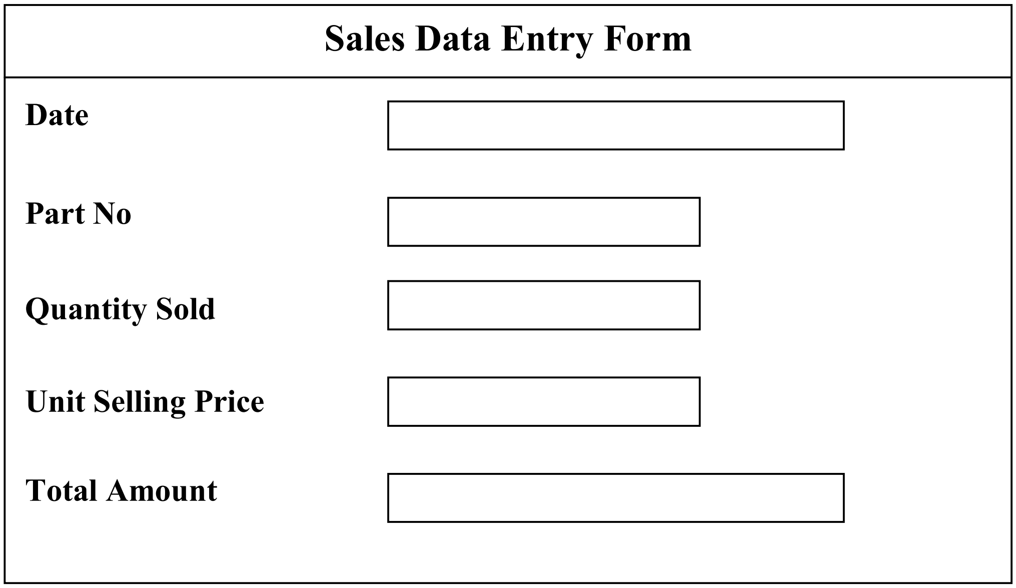
Blessed Superstores Ltd

Purchase Report from…………………To………….…………….
Report Printed on………….………..

Date	Part No.	Item Type	Item Name	Qty. Pur.	Unit Cost	Amount
dd/mm/yyyy						
dd/mm/yyyy	9999	xxx…..x	xxx…..x	9999	$$$$$.$	$$$$.$
dd/mm/yyyy						$$$$.$
dd/mm/yyyy						
dd/mm/yyyy	9999	xxx…..x	xxx…..x	9999	$$$$$.$	$$$$.$
dd/mm/yyyy	9999	xxx…..x	xxx…..x	9999	$$$$$.$	$$$$.$
dd/mm/yyyy						$$$$.$
Grand Total						$$$$.$

Figure C.5: Layout of periodic purchase report

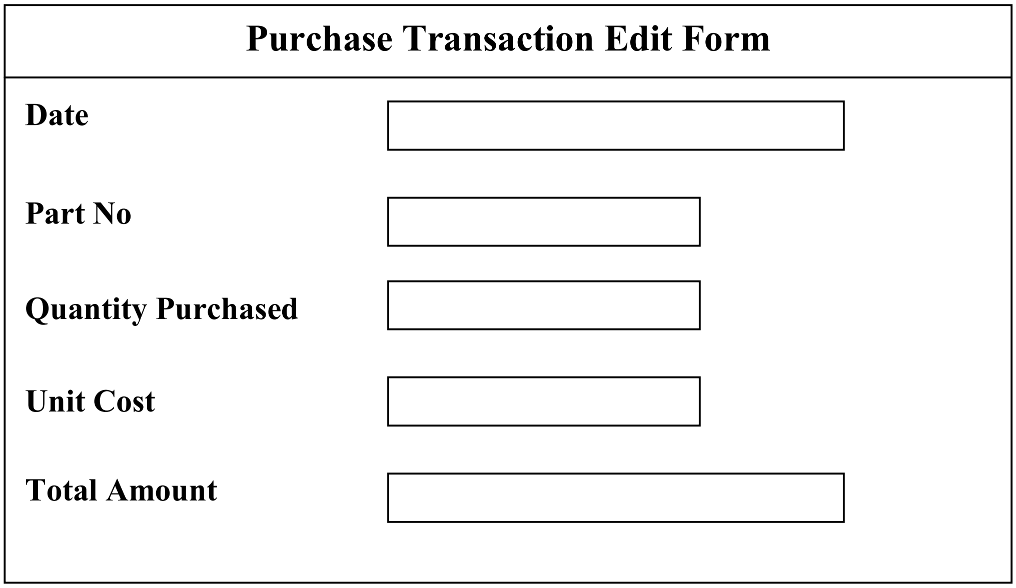
APPENDIX D: FORMATS OF THE COMPUTER INPUT DESIGNS



Sales Data Entry Form

Date	
Part No	
Quantity Sold	
Unit Selling Price	
Total Amount	

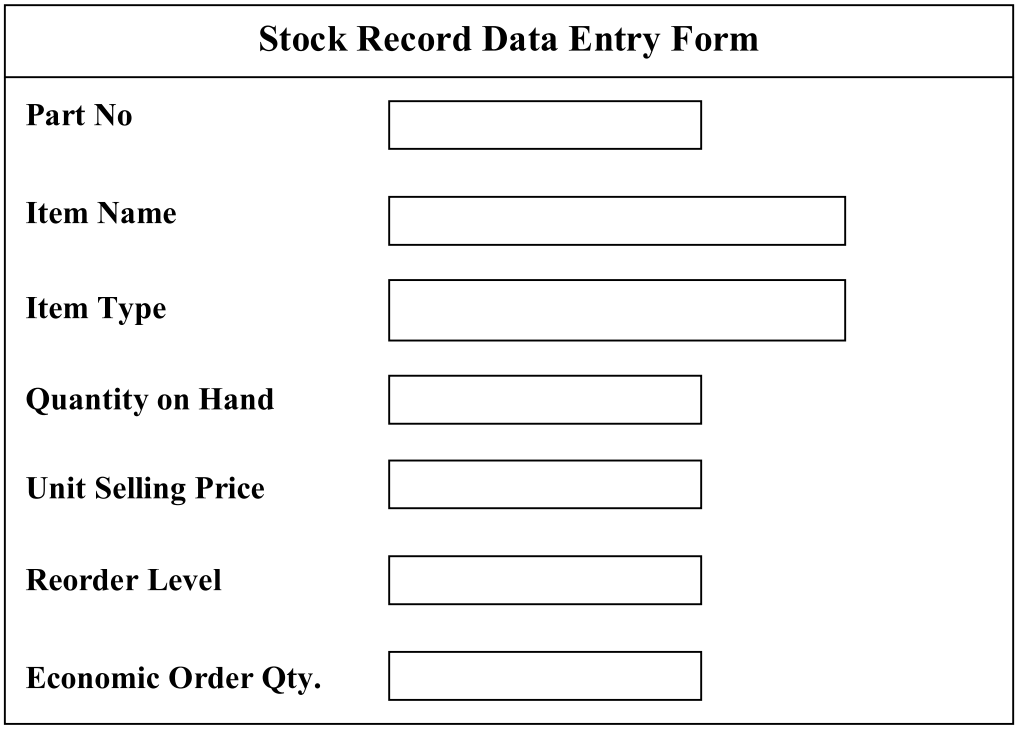
Figure D.1: Format of sales data input form



Purchase Transaction Edit Form

Date	
Part No	
Quantity Purchased	
Unit Cost	
Total Amount	

Figure D.2: Format of purchase transaction input form



Stock Record Data Entry Form

Part No

Item Name

Item Type

Quantity on Hand

Unit Selling Price

Reorder Level

Economic Order Qty.

Figure D.3: Format of stock record data input form

APPENDIX D

PROGRAM SOURCE CODES

□

```
‘Blessed Superstores Project Module
Type=Exe
Reference=*\G{00020430-0000-0000-C000-
000000000046}#2.0#0#C:\WINDOWS.000\SYSTEM\stdole2.tlb#OLE
Automation
Object={F9043C88-F6F2-101A-A3C9-08002B2F49FB}#1.2#0; COMDLG32.0CX
Object={6B7E6392-850A-101B-AFC0-4210102A8DA7}#1.3#0; COMCTL32.0CX
Reference=*\G{00025E01-0000-0000-C000-
000000000046}#4.0#0#C:\PROGRAM FILES\COMMON FILES\MICROSOFT
SHARED\DAO\DAO350.DLL#Microsoft DAO 3.5 Object Library
Object={00028C01-0000-0000-0000-000000000046}#1.0#0; DBGRID32.OCX
Object={0BA686C6-F7D3-101A-993E-0000C0EF6F5E}#1.0#0;THREED32.OCX
Object={A8B3B723-0B5A-101B-B22E-00AA0037B2FC}#1.0#0; GRID32.OCX
Object={00025600-0000-0000-C000-000000000046}#4.6#0; CRYSTL32.OCX
Object={FAEEE763-117E-101B-8933-08002B2F4F5A}#1.1#0; DBLIST32.OCX
Object={FE0065C0-1B7B-11CF-9D53-00AA003C9CB6}#1.1#0;
 COMCT232.OCX
Reference=*\G{81CCOA9A-1E1D-11D0-BAE1-
00A0D1001A5A}#4.0#0#C:\PROGRAM
FILES\DEVSTUDIO\VB\WIZARDS\DTADDIN.DLC#Microsoft Visual DataTools
AddIn for Visual Basic
Object={6FBA474E-43AC-11CE-9A0E-00AA0062BB4C}#1.0#0; SYSINFO.OCX
Object={5E9E78A0-531B-11CF-91F6-C2863C385E30}#1.0#0;
MSFLXGRD.OCX
Module= Module1; Blessed SuperstoresModule.bas
Form= frmMain.frm
```

```
Form= frmSplash.frm

Form=frmLogin.frm

Form=frmAbout.frm

Form=frmDataGrid.frm

Form=frmPURCHASE.frm

Form=frmSALES.frm

Form=frmPURCHASEGrid.frm

Form=frmSALESGrid.frm

Form=frmPassword.frm

Form=frmTopForm.frm

Form=frmBackUpRestore.frm

Form=frmAboutAuthor.frm

Form=frmCleanUp.frm

Form=frmMaintainOpts.frm

Form=frmQntyOnHand.frm

Form=frmItemsProperties.frm

Form=frmSalesReport.frm

Form=frmStockLevReport.frm

Form=frmStockRec.frm

Form=frmPurchaseReport.frm

Form=frmViewLogin.frm

Form=frmAboutBlessedSuperStores.frm

IconForm= “frmStoresMain”

Startup= “Sub Main”

HelpFile= “”

Title= “Blessed Superstores Limited Information System”
```

```
Command32= “vb5/ make Blessed Superstores”
Name= “Blessed Superstores Limited”
HelpContentID= “0”
Description= “Blessed Superstores Limited Information System Project”
CompatibleMode= “0”
MajorVer=1
MinorVer=0
RevisionVer=0
AutoIncrementVer=0
ServerSupportFiles=0
VersionCompanyName= “Grace Software Company, Umuahia”
CompilationType=0
OptimizationType=0
FavorPentiumPro(tm)=0
CodeViewDebugInfo=0
NoAliasing=0
BoundsCheck=0
OverflowCheck=0
FIPointCheck=0
FDIVCheck=0
UnroundedFP=0
StartMode=0
Unattended=0
ThreadPerObject=0
MaxNumberOfThreads=1
```

```
BlessedSuperStoresModule Module
Attribute VB_Name = “Module1”
Option Explicit

Global gsDatabase As String
Global gsConnect As String
Global gsRecordsource As String
Global To_Go_On As Boolean

Public fMainTopForm As Form1


Sub Main()
  Dim Timer1 As Timer


  frmLogin. Show vbModal
  If Not frmLogin.OK  Then
      ‘Login Failed so exit app
      End
   End If
   To_Go_On = False
   Unload frmLogin

   frmSplash.Show vbModal
   ‘frmSplash.Refresh

   ‘Set fMainForm= New frmMain
   ‘Load fMainForm

    Set fMainTopForm=New Form1
    Load fMainTopForm
    Unload frmSplash

    fMainTopForm.Show  ‘VbModal
End Sub

  Private Sub Timer1_Timer()
     Unload frmSplash
End Sub
```

```
‘frmTopForm Form Module
VERSION 5.00
Object              =       “{0BA686C6-F7D3-101A-993E-0000C0EF6F5E}#1.0#0”;
“THREED32.OCX”
Begin VB.Form Form1
   Caption        =    “Blessed Superstores Information System Main Form”
   ClientHeight   =     4395
   ClientLeft     =      210
   ClientTop      =      2070
   ClientWidth    =      6960
   LinkTopic      =     “ Form1”
   ScaleHeight    =      6795
   ScaleWidth     =      9480
   WindowState  =      2 ‘Maximized
   Begin Threed.SSCommand cmdAuthor
      Height       =      1215
      Left         =       5040
      TabIndex   =        13
      Top          =        5280
      Width        =       3975
      _Version    =       65538
      _ExtentX    =       7011
      _ExtentY    =       2143
      _StockProps  =      78
      Caption        =     “&Chukwu  J. N-(20074616268)”
      ForeColor     =      16777215
      BeginProperty Font{0BE35203-8F91-11CE-9DE3-00AA004BB851}
         Name         =       “Times New Roman”
         Size          =        14.25
         Charset      =         0
         Weight       =         700
         Underline  =          0 ‘False
         Italic         =          0 ‘False
         Strikethrough   =    0 ‘False
```

```
  EndProperty
  Font3D           =   2
End
Begin VB.Frame Frame4
  Caption          =  “Developer Info”
  BeginProperty Font
    Name           = “Times New Roman”
    Size           =   11.25
    Charset        =    0
    Weight         =   700
    Underline      =   0 ‘False
    Italic         =   0 ‘False
    Strikethrough  =   0  ‘False
  EndProperty
  ForeColor        =    &H0000FF00&
  Height           =     1575
  Left             =     4920
  TabIndex         =      12
  Top              =      5040
  Width            =      4215
 End
 Begin VB.Timer Timer1
   Interval        =     60
   Left            =     4920
   Top             =     2040
 End
 Begin Threed.SSPanel SSPanelTime
   Height          =      975
   Left            =      5880
   TabIndex        =         11
   Top             =      1920
   Width           =      2655
   _Version        =       65536
   _ ExtentX       =       4683
 _ExtentY        =        1720
 _StockProps     =        15
 ForeColor       =         16777215
 BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
    Name           =       "Times New Roman"
    Size           =       18
    Charset        =        0
    Weight         =        700
    Underline      =          0 'False
    Italic         =       0 'False
    Strikethrough  =       0 'False
  EndProperty
  BevelWidth       =       3
  BorderWidth      =       5
  BevelOuter      =       1
  BevelInner      =       2
  FloodColor       =      32768
  Font3D           =      2
End
Begin Threed.SSCommand cmdQuit
 Height          =       735
 Left            =        7200
 TabIndex        =          10
 Top             =       3600
 Width           =        1455
 _Version        =        65536
 _ ExtentX       =        2566
 _ExtentY        =        1296
 _StockProps     =          78
 Caption         =         "&Quit"
 ForeColor       =        16776960
 BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
    Name           =      "Times New Roman"
    Size           =       14.25
```

```
      Charset       =     0
      Weight        =     700
      Underline     =     0 ‘False
      Italic        =     0 ‘False
      Strikethrough =     0 ‘ False
    EndProperty
    BevelWidth      =      4
    Font3D          =      2
    Picture         =      “frmTopForm.frx”:0000
  End
  Begin Threed.SSCommand cmdAbout
    Height          =     735
    Left            =     5400
    TabIndex        =       9
    Top             =     3600
    Width           =     1455
    _Version        =     65536
    _ ExtentX       =     2566
    _ExtentY        =     1296
    _StockProps     =       78
    Caption         =     “A&bout…”
    ForeColor       =     16776960
    BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
      Name          =     “Times New Roman”
      Size          =     14.25
      Charset       =     0
      Weight        =     700
      Underline     =     0 ‘False
      Italic        =     0 ‘False
      Strikethrough =     0 ‘False
    EndProperty
    BevelWidth      =      4
    Font3D          =      2
End
```

```
Begin Threed.SSPanel SSPanel5
   Height          =      1695
   Left            =       5040
   TabIndex         =         8
   Top             =      3120
   Width          =       4095
   _Version        =        65536
   _ ExtentX        =        7223
   _ExtentY         =        2990
   _StockProps     =          15
   ForeColor        =        16777215
   BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
      Name           =      “MS Sans Serif”
      Size             =       8.25
      Charset          =        0
      Weight            =        400
      Underline        =      0 ‘False
      Italic              =      0 ‘False
      Strikethrough   =      0 ‘False
    EndProperty
    BevelWidth        =      3
    BorderWidth       =      4
    BevelOuter       =      1
    BevelInner        =      2
    FloodColor        =      32768
 End
 Begin  VB.Frame Frame3
   Height          =      2055
   Left            =       4920
   TabIndex         =         7
   Top             =      2880
   Width          =       4335
End
Begin Threed.SSPanel SSPanAction
```

```
  Height          =      1335
  Left            =      5040
  TabIndex        =         6
  Top             =       360
  Width           =      4095
  _Version        =     65536
  _ ExtentX       =      7223
  _ExtentY        =      2355
  _StockProps     =        15
  ForeColor       =     65535
  BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
     Name           =     “Times New Roman”
     Size           =      11.25
     Charset        =       0
     Weight         =      700
     Underline      =     0 ‘False
     Italic         =     0 ‘False
     Strikethrough  =     0 ‘False
   EndProperty
   BevelWidth       =    2
   BevelOuter       =    1
   BevelInner       =    1
   Outline          =   -1 ‘True
   FloodColor       =     32768
End
Begin  VB.Frame Frame2
   Caption               =       “Action”
   BeginProperty Font
     Name          =  “Times New Roman”
     Size          =   12
     Charset       =    0
     Weight        =   700
     Underline     =   0  ‘False
     Italic        =   0 ‘False
```

```
    Strikethrough  =    0  ‘False
  EndProperty
  ForeColor        =    &H0000FF00&
  Height           =     1815
  Left             =     4920
  TabIndex         =      5
  Top              =     0
  Width            =      4335
End
Begin Threed.SSCommand cmdStores
  Height           =     855
  Left             =     1560
  TabIndex         =       4
  Top              =      600
  Width            =      1815
  _Version         =      65536
  _ ExtentX        =      3201
  _ExtentY         =       1508
  _StockProps      =        78
  Caption          =     “&Stores”
  ForeColor        =      16711680
  BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
    Name           =    “Times New Roman”
    Size           =     18
    Charset        =      0
    Weight         =     700
    Underline      =    0 ‘False
    Italic         =    0 ‘False
    Strikethrough  =    0  ‘False
  EndProperty
  Font3D           =      2
  MouseIcon        =      “frmTopForm.frx”:001C
  Picture          =      “frmTopForm.frx”:0038
End
```

```
Begin Threed.SSCommand cmdAccounts
   Height          =        855
   Left            =        480
   TabIndex        =         14
   Top             =        240
   Width           =         1935
   _Version        =        65536
   _ ExtentX       =         3413
   _ExtentY        =         1508
   _StockProps     =          78
   Caption         =         "&Accounts"
   ForeColor       =         16711935
   BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
     Name          =      "Times New Roman"
     Size          =       14.25
     Charset       =        0
     Weight        =       700
     Underline     =        0 'False
     Italic        =       0 'False
     Strikethrough  =      0 'False
   EndProperty
   Font3D           =       2
   MouseIcon        =       "frmTopForm.frx":15DC
   Picture          =       "frmTopForm.frx":15F8
  End
End
Begin Threed.SSPanel SSPanelAccounts
   Height          =        1335
   Left            =        960
  TabIndex         =         2
  Top              =        1920
  Width            =         2895
  _Version         =        65536
  _ ExtentX        =         5106
  _ExtentY        =         2355
  _StockProps     =          15
  BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
     Name           =      "Times New Roman"
     Size           =       9.75
     Charset        =        0
     Weight         =       700
     Underline      =       0 'False
     Italic         =       0 'False
     Strikethrough  =       0 'False
  EndProperty
  BevelWidth      =       2
  BevelInner      =      2
  FloodColor      =       32768
  Font3D          =        1
 Begin  VB.Frame Frame1
  Caption               =    "Application SubSystems"
  BeginProperty Font
     Name           = "Times New Roman"
     Size          =    11.25
     Charset        =    0
     Weight         =    700
     Underline      =    0 'False
     Italic         =    0 'False
     Strikethrough  =    0  'False
   EndProperty
   ForeColor        =    &H0000FF00&
   Height          =       6615
   Left            =       120
  TabIndex        =         0
  Top            =         0
  Width          =        4575
Begin Threed.SSPanel SSPanelStores
  Height          =       1335
  Left            =       840
  TabIndex        =        1
  Top             =        360
  Width           =        2895
  _Version        =        65536
  _ ExtentX       =        5106
  _ExtentY        =         2355
  _StockProps     =          15
  BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
     Name          =      "Times New Roman"
     Size          =       9.75
     Charset       =        0
     Weight        =       700
     Underline     =       0 'False
     Italic        =       0 'False
     Strikethrough  =      0 'False
  EndProperty
  BevelWidth       =      2
  BevelInner       =     2
  FloodColor       =      32768
  Font3D           =       1
End
Begin Threed.SSPanel SSPanelMaintain
  Height          =       1335
  Left            =       840
  TabIndex        =        16
  Top             =       5040
  Width           =        2895
  _Version        =        65536
  _ ExtentX       =        5106
  _ExtentY        =         2355
  _StockProps     =          15
  BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
     Name          =      "Times New Roman"
```

```
      Size            =      9.75
      Charset         =       0
      Weight          =      700
      Underline       =      0 ‘False
      Italic          =      0 ‘False
      Strikethrough   =      0 ‘False
   EndProperty
   BevelWidth      =      2
   BevelInner      =      2
   FloodColor      =      32768
   Font3D          =       1
Begin Threed.SSCommand cmdMaintain
   Height          =      855
   Left            =      360
   TabIndex        =       17
   Top             =       240
   Width           =      2175
   _Version        =      65536
   _ ExtentX       =       3836
   _ExtentY        =       1508
   _StockProps     =        78
   Caption         =       “&Maintenance”
   ForeColor       =        16711680
   BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
      Name            =      “Times New Roman”
      Size            =      14.25
      Charset         =       0
      Weight          =      700
      Underline       =      0 ‘False
      Italic          =      0 ‘False
      Strikethrough   =      0 ‘False
   EndProperty
   Font3D          =       2
   MouseIcon       =      “frmTopForm.frx”:2F4A
```

```
           Picture               =        “frmTopForm.frx”:2F66
          End
         End
       End
       Begin  VB.Image Image3
           Height           =        495
           Left               =        480
          Top              =        2760
          Width           =         1215
       End
       Begin  VB.Image Image1
           Height           =        495
           Left               =        480
          Top              =        2760
          Width           =         1215
       End
End
Attribute VB_Name  = “Form1”
Attribute VB_GlobalNameSpace  = False
Attribute VB_Creatable  = False
Attribute VB_Predeclaredld  = True
Attribute VB_Exposed  = False
Option Explicit

Private Sub cmdAbout_Click()
   Dim f As New frmAbout
   f.Show vbModal, Me
End Sub

Private Sub cmdAbout_GotFocus()
    SSPanAction.Caption = “Shows.information about the application”
End Sub

Private Sub cmdAbout_LostFocus()
```

```
  SSPanAction.Caption = “”
End Sub

Private Sub cmdAbout_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    cmdAbout.SetFocus
End Sub

Private Sub cmdAccounts_Click()
   Dim f As New frmAccountsInfo
   Beep
   f.Show vbModal, Me
End Sub

Private Sub cmdAccounts_GotFocus()
   cmdAccounts.BevelWidth = 4
   SSPanAction.Caption =  “Handles all accounting tasks”
End Sub

Private Sub cmdAccounts_LostFocus()
  cmdAccounts.BevelWidth = 2
  SSPanAction.Caption = “”
End Sub

Private Sub cmdAccounts_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
   ‘cmdAccounts.BevelWidth=4
    cmd Accounts.SetFocus
End Sub

Private Sub cmdAuthor_Click()
   Dim f As New frmAboutAuthor
   f.Show vbModal, Me
End Sub

Private Sub cmdAuthor_GotFocus()
```

```
   cmdAuthor.ForeColor=QBColor(15) ‘bright white
End Sub

Private Sub cmdAuthor_LostFocus()
   cmdAuthor.ForeColor=QBColor(7) ‘white/gray
End Sub

Private Sub cmdAuthor_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
   SSPanAction.Caption= “Shows Information about the Software Developer”
End Sub

Private Sub cmdMaintain_Click()
   Dim f As New frmMaintainOpts
      Me.Hide
      f.Show vbModal, Me
      Me.Show
End Sub

Private Sub cmdMaintain_GotFocus()
   cmdMaintain.BevelWidth = 4
   SSPanAction.Caption  = “Handles Password Administration,
Database Backup/Restore and Database Clean up”
End Sub

Private Sub cmdMaintain_LostFocus()
   cmdMaintain.BevelWidth = 2
   SSPanAction.Caption =  “”
End Sub

Private Sub cmdMaintain_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    cmdMaintain.SetFocus
End Sub

Private Sub cmdQuit_Click()
```

```
    Dim i As Integer, TimeNow As String

    ‘inserts the user logout time into the APPLOG Table
    TimeNow = Time$
    frmLogin.datLoginTable. Refresh
    frmLogin.datLoginTable. Recordset.MoveLast
    frmLogin.datLoginTable. Recordset.Edit
    frmLogin.datLoginTable. Recordset.Fields(“LogoutTime”) = TimeNow
    frmLogin.datLoginTable. Recordset.Update

    ‘unload all forms and end application
     For i = Forms. Count -1 To 1 Step -1
         Unload Forms(i)
     Next
     Unload Me
     End
 End Sub

Private Sub cmdQuit_GotFocus()
      SSPanAction.Caption =  “Quits The Application”
End Sub

Private Sub cmdQuit_LostFocus()
      SSPanAction.Caption =  “”
End Sub

Private Sub cmdQuit_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
    cmdQuit.SetFocus
End Sub

Private Sub cmdStores_Click()
   Dim fMainForm As New frmStoresMain
       frmStoresMain. Show ‘vbModal
End Sub
```

```
Private Sub cmdStores_GotFocus()
   cmdStores.BevelWidth = 4
   SSPanAction.Caption  = “Handles sales , purchases, and stock control related functions”
End Sub

Private Sub cmdStores_LostFocus()
    cmdStores.BevelWidth = 2
    SSPanAction.Caption = “”
End Sub

Private Sub cmdStores_MouseMove (Button As Integer. Shift As Integer, X As Single, Y As Single)
    ‘cmdStores.BevelWidth=4
     cmdStores.SetFocus
End Sub

Private Sub Form _Load()
   If frmLogin.AccessLevel = 1 Then Exit Sub
   cmdStores.Enabled = False
   cmdMaintain.Enabled = False
   If  frmLogin. UserUnit = "STR" Then cmdStores. Enabled = True
   If  frmLogin. UserUnit = "SLE" Then cmdStores. Enabled = True
   If  frmLogin. UserUnit = "CPS" Then cmdMaintain. Enabled = True
End Sub

Private Sub Form _Unload(Cancel As Integer)

    Call cmdQuit_Click
End Sub

Private Sub Frame4_MouseMove (Button As Integer. Shift As Integer, X As Single, Y As Single)
   SSPanAction.Caption = “”
End Sub
```

```
Private Sub SSPanel5_MouseMove (Button As Integer. Shift As Integer, X As Single, Y As Single)
   cmdAuthor.SetFocus
End Sub

Private Sub SSPanelAccounts_MouseMove (Button As Integer. Shift As Integer, X As Single, Y As
Single)
   cmdAuthor.SetFocus
End Sub

Private Sub SSPanelMaintain_MouseMove (Button As Integer. Shift As Integer, X As Single, Y As
Single)
   cmdAuthor.SetFocus
End Sub

Private Sub SSPanelStores_MouseMove (Button As Integer. Shift As Integer, X As Single, Y As Single)
   cmdAuthor.SetFocus
End Sub

Private Sub Timer1_Timer()
   SSPanelTime.Caption = Time$
End Sub
```

```
‘frmLogin Form Module
VERSION 5.00
Begin VB.Form frmLogin
    BorderStyle      =   3 'Fixed Dialog
    Caption          =  "Login Form"
    ClientHeight     =    2190
    ClientLeft       =     45
    ClientTop        =    2040
    ClientWidth      =     5460
    LinkTopic        =   " Form1"
    MaxButton        =    0 'False
    MinButton        =    0 'False
    ScaleHeight      =    2190
    ScaleWidth       =    5460
    ShowInTaskbar   =  0 'False
    Tag              =   "Login"
    Begin VB.Data datLoginTable
       Caption       =    "Data1"
       Connect       =     "dBASE IV;"
       DatabaseName    =    "C:\Blessed SuperStores"
       DefaultCursorType     =   0 'DefaultCursor
       DefaultType           =     2  'UseODBC
       Exclusive             =       0   ‘False
       Height                =      345
       Left                  =       2880
      Options                =         0
      ReadOnly           =         0 'False
      RecordSetType    =        0 ' Table
      RecordSource       =        "Applog"
      Top                =          0
      Visible            =         0 'False
     Width               =         1500
  End
  Begin VB.Data datPrimaryRS
```

```
     Caption              =    " Data1"
     Connect              =    "Paradox 5.x;"
     DatabaseName     =    "C:\Blessed SuperStores"
     DefaultCursorType     =   0 'DefaultCursor
     DefaulTtype           =         2  'UseJet
     Height               =        345
     Left                 =         3240
     Options                =         0
     ReadOnly           =         0 'False
     RecordSetType    =        0 ' Table
     RecordSource       =        "Password"
     Top                  =          600
     Visible              =        0 'False
     Width                =        1140
 End
 Begin VB.CommandButton cmdCancel
     Cancel               =         -1  'True
     Caption               =        “&Cancel”
     BeginProperty Font
         Name           = “FTG-Normal”
          Size             =    12
          Charset        =    0
          Weight          =    700
          Underline      =    0 ‘False
          Italic            =    0 ‘False
          Strikethrough  =    0 ‘False
       EndProperty
       Height            =        480
       Left               =        3360
       TabIndex         =         2
       Tag               =         " Cancel"
       Top               =          1560
       Width            =          1140
End
```

```
Begin VB.CommandButton cmdOK
   Caption          =      “&OK”
   Default          =      -1  ‘True
   BeginProperty Font
      Name          = “FTG-Normal”
      Size          =   12
      Charset       =    0
      Weight        =   700
      Underline     =   0 ‘False
      Italic        =   0 ‘False
      Strikethrough =   0 ‘False
   EndProperty
   Height           =      480
   Left             =      1080
   TabIndex         =       1
   Tag              =      " OK"
   Top              =      1560
   Width            =      1140
End
Begin VB.TextBox txtPassword
   BeginProperty Font
      Name          = “FTG-Normal”
      Size          =   11.25
      Charset       =    0
      Weight        =   700
      Underline     =   0 ‘False
      Italic        =   0 ‘False
      Strikethrough =   0 ‘False
   EndProperty
   Height           =     360
   IMEMode          =     3 'DISABLE
   Left             =      1800
   MaxLength        =       10
   PasswordChar     =      "*"
```

```
      TabIndex         =         0
      Top              =         360
      Width            =         1365
   End
   Begin VB.Label lbeUserName
       BorderStyle      =    1  'Fixed Single
       BeginProperty Font
         Name           = “FTG-Normal”
         Size           =    9.75
         Charset        =     0
         Weight         =    700
         Underline      =    0  ‘False
         Italic         =   1 ‘True
         Strikethrough  =    0 ‘False
       EndProperty
       ForeColor        =        &H00FFFFFF&
       Height           =         375
       Left             =          1800
       TabIndex         =          5
       Top              =           960
       Width            =           3495
   End
   Begin VB. Label lblLables
       AutoSize         =    -1  'True
       BorderStyle      =    1  'Fixed Single
       Caption          =    "&Password:"
       BeginProperty Font
         Name           = “FTG-Normal”
         Size           =    9.75
         Charset        =     0
         Weight         =    700
         Underline      =    0 ‘False
         Italic         =   0  ‘False
         Strikethrough  =    0 ‘False
```

```
    EndProperty
    ForeColor       =       &H0000FFFF&
    Height          =         315
    Index           =          1
    Left            =          240
    TabIndex        =          3
    Tag             =        "&Password:"
    Top             =          360
    Width           =          1245
End
Begin VB. Label lblLables
    AutoSize         =    -1  'True
    BorderStyle      =    1  'Fixed Single
    Caption          =    "&UserName:"
    BeginProperty Font
      Name           = "FTG-Normal"
      Size           =    9.75
      Charset        =     0
      Weight         =    700
      Underline      =    0 'False
      Italic         =   0 'False
      Strikethrough  =    0 'False
    EndProperty
    ForeColor       =       &H0000FFFF&
    Height          =         315
    Index           =          1
    Left            =          240
    TabIndex        =         4
    Tag             =        "&UserName:"
    Top             =          960
    Width           =          1380
   End
 End
 Attribute VB_Name  = "frmLogin"
 Attribute VB_GlobalNameSpace  = False
 Attribute VB_Creatable  = False
 Attribute VB_Predeclaredld  = True
 Attribute VB_Exposed  = False
 Option Explicit

 'Private Declare Function  GetUserName  Lib "advapi32.dll" Alias
 "GetUserNameA" (ByVal Ipbuffer As String, nSize As Long) As long
 Public  OK As Boolean AccessLevel As Integer, UserUnit As String
 Dim Called_PassLostFocus  As Boolean, cycle As Integer, Pass$, ch$, i%

 Private Function CodePassword$(X$)
    Pass$ = ""
    For i% = 1 To Len(X$)
        ch$  =  Mid$(X$, i%, 1)
       Pass$ = Pass$ + Chr$(Asc(ch$) + 2)
    Next
    CodePassword$ = Pass$
End Function

Private Sub Form_Load()
    'Dim sBuffer As String
    'Dim ISize As Long
    'sBuffer   =  Space$(255)
    'ISize      =  Len(sBuffer)
    'Call GetUserName(sBuffer,ISize)
    'If ISize > 0 Then
    'lbeUserName.Caption = Left$(sBuffer, ISize)
    ‘Else
    'lbeUserName.Caption = vbNullString
    ‘End If
    cycle   =  0
    datLoginTable. Refresh
    datPrimaryRS.Refresh
```

```
  datPrimaryRS.Recordset.Index  =  "Password#Px"
  Called_PassLostFocus  = False
   lbeUserName.Caption = "    "
End Sub

Private Sub cmdCancel_Click()
   OK =  False
   Me. Hide
End Sub

Private Sub cmdOK_Click
   Dim Dept As String
   If Called_PassLostFocus  = False Then Call txtPassword_LostFocus
   Called_PassLostFocus  = False
   cycle    =    cycle   +   1
   With datPrimaryRS.Recordset      'password table recordset
        'Criteria  = "UserId  = ""& Trim(CodePassword$(txtPassword.Text))&""
       .FindFirst Criteria
       .Seek  "=" , Trim(CodePassword$(txtPassword .Text))
       If  .NoMatch = False Then  ' correct password
             OK    =   True
              AccessLevel = datPrimaryRS. Recordset Fields("AccessLevel"). Value
              Dept = datPrimaryRS. Recordset Fields("Department") .Value
              UserUnit = Left$(Dept, 3)
              Me.Hide
              GoSub LoginData
              Exit Sub
       Else
              If cycle < 3 Then
                MsgBox  "Invalid Password, try again !",  ," Login"
              Else
                MsgBox  "Invalid Password, No More trials!"
              End If
              If cycle = 3 Then
```

```
                  Me.Hide: Exit Sub
                 End If
                 txtPassword. SetFocus
                 txtPassword. SelStart  = 0
                 txtPassword. SelLength  = Len(txtPassword.Text)
            End  if
         End With
         Exit Sub

Login Data
     Dim TimeNow As String, DateNow As Date
     TimeNow = Time$
     DateNow =  Date$

    datLoginTable.Refresh
    With datLoginTable.Recordset. Add New
        .Fields("Username")  = Trim(lbeUserName.Caption)
        .Fields("Password")   = CodePassword$(txtPassword.Text)
        .Fields("LoginTime") = TimeNow
        .Fields("LogDate")    =  DateNow
        .Fields("Department") = Dept
    End With
    Return
EndSub

Private Sub txtPassword_LostFocus()
    Called_PassLostFocus  =  True
    datPrimaryRS.Recordset.Seek "=" , CodePassword$(txtPassword.Text)
    If datPrimaryRS.Recordset.NoMatch  = False Then
        lbeUserName.Caption  = datPrimaryRS.Recordset. Fields("UserName")
    Else
        lbeUserName.Caption   =    "      "
    End If
End Sub
```

```
'frmMain Form module
VERSION 5.00
Object        =        "{6B7E6392-850A-101B-AFC0-42101012A8DA7}#1.1#0";
“COMCTL32.OCX”
Object        =        “{0BA686C6-F7D3-101A-993E-0000C0EF6F5E}#1.0#0”;
"THREED32.OCX"
Begin VB.Form frmStoresMain
    AutoRedraw      =   -1  'True
    Caption         =   "Blessed SuperStores Limited Information System – Stores Subsystem"
    ClientHeight    =    6495
    ClientLeft      =     240
    ClientTop       =    3075
    ClientWidth     =    9480
    ClipControls    =     0   'False
    Icon       =   " frmMain.frx":0000
    LinkTopic       =   "Form 1"
    ScaleHeight     =    6495
    ScaleWidth      =    9480
    WindowState     =     2   'Maximized
    Begin VB. PictureBox Picture 1
      Height      =       4335
      Left               =        1920
      Picture            =         " frmMain.frx":030A
      ScaleHeight        =        4275
      ScaleWidth         =          5835
      TabIndex           =          2
      Top                =          960
      Width              =          5895
  End
  Begin VB.Timer Timer1
      Interval           =          60
      Left               =          4200
      Top                =          3000
   End
   Begin Threed.SSPanel SSPanel1
      Height          =           6135
      Left            =             0
      TabIndex        =             1
      Top             =             0
      Width           =           9495
      _Version         =           65536
      _ExtentX         =           16748
      _ExtentY         =           10821
      _StockProps    =           15
      BeginProperty Font  {0BE35203-8F91-11CE-9DE3-00AA004BB851}
         Name            =         "MS Sans Serif"
         Size            =          8.25
         Charset         =          0
         Weight          =          400
         Underline       =          0 'False
         Italic          =          0 'False
         Strikethrough     =          0 'False
   EndProperty
   BevelWidth        =     3
   BorderWidth       =     8
   BevelOuter       =     1
   BevelInner       =     2
   FloodColor        =    32768
End
Begin ComctlLib. StatusBar sbStatusBar
   Align           =     2   'Align Bottom
   Height          =          285
   Left            =            0
   TabIndex        =            0
   Top             =          6210
   Width           =          9480
   _ExtentX        =          16722
   _ExtentY        =           503
```

```
 Simple Text       =              ""
 _Version          =          327680
 BeginProperty Panels {0713E89E-850A-101B-AFC0-4210102A8DA7}
   NumPanels      =    3
   BeginProperty Panel1 {0713E89F-850A-101B-AFC0-4210102A8DA7}
      AutoSize      =    1
      Object.Width     =     11165
      Text              =      "Stores Processing Subsystem"
      TextSave         =       "Stores Processing Subsystem"
      Key               =        ""
      Object.Tag         =        ""
      Object.ToolTipText    =   ""
   EndProperty
   BeginPropertyPanel2  {0713E89F-850A-101B-AFC0-4210102A8DA7}
      Style             =    6
      Autosize          =    2
      Textsave          =    02/08/2010
      Key                =    ""
      Object .Tag        =    ""
   EndProperty
   BeginPropertyPanel3  {0713E89F-850A-101B-AFC0-4210102A8DA7}
      Autosize          =      2
      Textsave          =   “”
      Key                =    ""
      Object. Tag        =    ""
   EndProperty
EndProperty
BeginProperty Font     {0BE35203-8F91-11CE-9DE3-00AA004BB851}
   Name           = “FTG-Normal”
   Size           =     9
   Charset        =    0
   Weight          =   700
    Underline      =    0 ‘False
    Italic          =    0 ‘False
```

```
      Strikethrough  =   0 ‘False
  EndProperty
  MouseIcon         =     "frmMain.frx":07F2
End
Begin VB. Menu mnuFile
    Caption             =    "&File "
    Begin VB. Menu mnuFileNewStock
        Caption            =    "&New Stock  "
    End
    Begin VB. Menu mnuFileBar1
        Caption            =      "_ "
    End
    Begin VB. Menu mnuFileTransact
        Caption              =     "&Transaction  Processing  "
    Begin VB. Menu mnuFileTransactSales
        Caption              =      "&Sales  "
        Shortcut             =       ^S
    End
    Begin VB. Menu mnuTransBar1
        Caption              =      "_"
        End
    Begin VB. Menu mnuTransactPurch
        Caption              =      "P&urchases "
        Shortcut             =       ^U
    End
End
Begin VB. Menu mnuFileBar2
    Caption        =    "_"
End
Begin VB. Menu mnuFileExit
    Caption        =  "E&xit  "
 End
End
Begin VB. Menu mnuView
```

```
  Caption            =      "&View  "
Begin VB. Menu mnuViewData
   Caption        =    "&DataForms  "
Begin VB. Menu mnuViewDataSTOCKS
   Caption    =      "Stocks  "
End
Begin VB. Menu mnuViewSales
   Caption    =      "Sales "
End
Begin VB .Menu mnuViewPurchase
   Caption       =     "Purchase  "
  End
End
Begin VB. Menu mnuViewBar5
   Caption            =      "_"
End
Begin VB. Menu mnuViewBar4
   Caption            =      "_  "
End
Begin VB. Menu mnuViewStatusBar
   Caption           =      "Status &Bar  "
   Checked          =          -1  'True
End
Begin VB. Menu mnuViewBar6
   Caption          =      "_  "
End
Begin VB. Menu mnuFileBar3
   Caption         =      "_  "
 Begin VB. Menu mnuViewRefresh
    Caption          =      "&Refresh "
   End
 End
 Begin VB. Menu mnuQuery
    Caption           =      "&Query  "
    Begin VB. Menu mnuQueryStockProperty
        Caption          =      "S&tockProperties "
        Begin VB. Menu mnuQryGeneral
           Caption          =      "&General  Properties "
        End
        Begin VB. Menu mnuQryBar2
           Caption            =      "_ "
        End
        Begin VB. Menu mnuQryByItems
           Caption           =      "&ByItems "
        End
    End
End
Begin VB. Menu mnuReport
    Caption            =      "&Report  "
    Begin VB. Menu mnuReportSales
        Caption            =      "&SalesReports... "
    End
    Begin VB. Menu mnuReportBar2
        Caption            =      "_  "
    End
    Begin VB. Menu mnuReportPurchase
        Caption            =      "&Purchases.....  "
    End
    Begin VB. Menu mnuReportBar3
        Caption            =      "_  "
    End
    Begin VB. Menu mnuReportStockLevel
        Caption          =      "&StockLevel "
    End
End
Begin VB. Menu mnuHelp
    Caption            =      "&Help  "
     Begin VB. Menu mnuHelpCompany
```

```
        Caption             =      "About Blessed SuperStores&Limited "
      End
      Begin VB. Menu mnuHelpBar1
        Caption             =      "_ "
      End
      Begin VB. Menu mnuHelpAbout
        Caption             =      "&AboutSoftware…  "
      End
  End
End
Attribute VB_Name  = “frmStoresMain”
Attribute VB_GlobalNameSpace  = False
Attribute VB_Creatable  = False
Attribute VB_Predeclaredld  = True
Attribute VB_Exposed  = False
Option Explicit
'Private Declare Function OSWinHelp% Lib "user32" Alias "WinHelpA" (By Val hwnd&, ByVal
HelpFile$, ByVal wCommand%. dwData As Any)
Private Sub Form_Load()
   Me.Left = GetSetting(App.Title, "Settings", " MainLeft", 1000)
   Me.Top = GetSetting(App.Title, "Settings", " MainTop", 1000)
   Me.Width= GetSetting(App.Title, "Settings", " MainWidth", 6500)
   Me.Height= GetSetting(App.Title, "Settings", " MainHeight ", 6500)
   If frmLogin. AccessLevel  =   3 Then
      mnuFileNewStock. Enabled   =   False
      mnuFileTransact. Enabled   =   False
   End If
End Sub
Private Sub Form_Unload(Cancel As Integer)
   Dim  i As Integer
   'close all sub forms
   For i =  Forms.Count - 1 To 1 Step -1
       Unload Forms(i)
   Next
```

```
    If Me. WindowState<>vbMinimized Then
        SaveSetting App.Title, 'Settings", "MainLeft", Me.Left
        SaveSetting App.Title, 'Settings", "MainTop", Me.Top
        SaveSetting App.Title, 'Settings", "MainWidth", Me.Width
        SaveSetting App.Title, 'Settings", "MainHeight", MeHeight
End Sub

Private Sub mnuReportStockLevel_Click()
    Dim f As New frmStockLevReport
    f.Show vbModal, Me
End Sub

Private Sub mnuQryByItems_Click()
    Dim f As New frmItemsProperties
    f.Show vbModal, Me
End Sub

Private Sub mnuQryGeneral_Click()
    Dim f As New frmStockProperties
    f.Show vbModal, Me
End Sub

Private Sub mnuReportPurchase_Click()
   Dim f As New frmPurchaseReport
   f.Show vbModal, Me
End Sub

Private Sub mnuReportSales_Click()
   Dim f As New frmSalesReport
   f.Show vbModal, Me
End Sub

Private Sub mnuTransactPurch_Click()
   Dim f As New frmPURCHASE
   f.Show vbModal, Me
End Sub

Private Sub mnuTransactSales_Click()
   Dim f As New frmSALES
   f.Show vbModal, Me
End Sub

Private Sub mnuViewPurchase_Click()
   frmPURCHASEGrid.Show vbModal, Me
End Sub

Private Sub mnuViewSales_Click()
   Dim f As New frmSALESGrid
   f.Show vbModal, Me
End Sub

Private Sub mnuFileNewStock_Click()
   Dim f As New frmStocksRec
   ‘If frmLogin.AccessLevel <> 3 Then
        f.Show vbModal, Me
   'Else
   'MsgBox  "You are not allowed to access this task!", vbInformation + vbOKOnly
   'End If
End  Sub

Private Sub mnuHelpCompany_Click()
   Dim f As New frmAboutBlessedSuperStoresLimited
   f.Show vbModal, Me
End Sub


Private Sub mnuViewDataSTOCKS_Click()
   Dim f As New frmDataGrid
   f.Show vbModal, Me
End Sub

Private Sub mnuHelpAbout_Click()
   frmAbout.Show vbModal, Me
End Sub

Private  Sub mnuViewStatusBar_Click()
   If mnuViewStatusBar.Checked Then
       sbStatusBar.Visible = False
       mnuViewStatusBar.Checked   = False
   Else
       sbStatusBar.Visible = True
       mnuViewStatusBar.Checked  = True
   End If
End Sub

Private Sub mnuViewRefresh_Click()
   Me.Refresh
End Sub

Private Sub mnuFileExit_Click()
   Me.Hide
End Sub

Private Sub Timer1_Timer()
   sbStatusBar.Panels(3).Text = Time$
End Sub
```

```
‘frmStocks Form Module
VERSION 5.00
Begin VB. Form frmSTOCKS
  BorderStyle      =   3 'Fixed Dialog
  Caption          =  " STOCKS "
  ClientHeight     =    4560
  ClientLeft       =     45
  ClientTop        =     345
  ClientWidth      =    7185
  LinkTopic        =  " Form1"
  MaxButton        =   0 'False
  MinButton        =   0 'False
  ScaleHeight      =   4560
  ScaleWidth       =    7185
  ShowInTaskbar  =  0  'False
  StartUpPosition  =  1  'CenterOwner
  Tag              =    " STOCKS "
  Begin VB.ListBox ListPartNo
    Height           =      675
    Left             =       1680
    TabIndex         =        17
    Top              =         0
    Visible          =        0  'False
    Width            =      1695
  End
  Begin VB.ComboBox Combo1
    DataField     =    "ItemType"
    DataSource     =  "datStocks"
    Height           =        315
    ItemData         =       "frmSTOCKS.frx":0000
    Left             =         3600
    List             =         "frmSTOCKS.frx":0016
    TabIndex       =          2
    Top            =         1080
    Width            =           1695
  End
  Begin VB. TextBox txtItemPartNo
    DataField     =    "PartNo"
    DataSource     =   "datStocks"
    Height           =         375
    Left               =         3600
    MaxLength      =          5
    TabIndex       =           2
    Top               =          120
    Width            =           1215
  End
  Begin VB. TextBox txtItemName
    DataField        =     "ItemName"
    DataSource      =    "datStocks"
    BeginProperty Font
      Name           =  “Times New Roman”
      Size             =    9.75
      Charset        =    0
      Weight         =   400
      Underline     =   0  ‘False
      Italic            =     0  ‘False
      Strikethrough  =   0  ‘False
    EndProperty
    Height           =        375
    Left               =        3600
    MaxLength      =         25
    TabIndex        =          1
    Top               =        600
    Width            =          3375
  End
  Begin VB. CommandButton cmdClose
    Caption       =     "&Close"
    BeginProperty Font
       Name            =  “Times New Roman”
       Size            =      14.25
       Charset         =     0
       Weight          =     700
       Underline       =      0  ‘False
        Italic           =     0  ‘False
        Strikethrough  =     0  ‘False
     EndProperty
     Height           =         420
     Left             =         5160
     TabIndex         =        10
     Tag              =         “&Close”
     Top              =          3720
     Width            =          975
  End
  Begin VB.CommandButton cmdDelete
    Caption       =      "&Delete"
    BeginProperty Font
       Name            =  “Times New Roman”
       Size            =      12
       Charset         =     0
       Weight          =     700
       Underline       =      0  ‘False
       Italic           =     0  ‘False
       Strikethrough  =     0  ‘False
    EndProperty
    Height            =         420
    Left              =       3240
    TabIndex          =          9
    Tag              =        '&Delete"
    Top               =          3720
    Width            =          975
  End
```

```
Begin VB.CommandButton cmdAdd
 Caption       =     "&Add"
 BeginProperty Font
  Name           =  “Times New Roman”
  Size           =      14.25
  Charset        =     0
  Weight         =     700
  Underline      =      0  ‘False
  Italic         =     0  ‘False
  Strikethrough  =    0  ‘False
 EndProperty
 Height          =       420
 Left            =       1200
 TabIndex        =         8
 Tag             =       '&Add"
 Top             =         3720
 Width           =         975
End
Begin VB.Data datStocks
  Align              =     2 ‘Align Bottom
  Caption             =     " STOCKS TABLE "
  Connect             =    "Paradox 5.x;"
  DatabaseName   =     "C:\BlessedSuperstores"
  DefaultCursorType  =   0   'DefaultCursor
  DefaultType          =       2   'UseODBC
  Exclusive            =     0  'False
  Height               =     345
  Left                 =      0
  Options               =      0
  ReadOnly            =       0   'False
  RecordsetType    =        0  'Table
  RecordSource     =        "STOCKS"
  Top                 =         4215
  Width                =        7185
 End
 Begin VB.TextBox txtFields
   DataField   =     "EconOrderQnty"
   DataSource  =    "datStocks"
   Height            =       375
   Index            =        4
   Left              =       3600
   TabIndex         =        6
   Top              =        3000
   Width             =       1935
 End
 Begin VB.TextBox txtFields
   DataField   =     "ReorderLevel"
   DataSource  =    "datStocks"
   Height            =       375
   Index            =        3
   Left              =      3600
   TabIndex         =        5
   Top              =       2520
   Width             =       1935
 End
 Begin VB.TextBox txtFields
   DataField   =     "UnitSellPrice"
   DataSource  =    "datStocks"
   Height            =       375
   Index            =        2
   Left              =        3600
   TabIndex         =        4
   Top              =       2040
   Width             =       1935
End
Begin VB.TextBox txtFields
    DataField   =     "QntyOnHand"
    DataSource  =    "datStocks"
   Height              =       375
   Index              =         4
   Left                =         1
   TabIndex          =         6
   Top                =        1560
   Width              =        1935
End
Begin VB.Label Label2
   Autosize      =    -1  'True
   Caption       =     "Item Type:"
   BeginProperty Font
      Name           =  “Times New Roman”
      Size            =     12
      Charset        =      0
      Weight         =      700
      Underline      =      0  ‘False
      Italic            =     -1  'True
      Strikethrough  =    0  ‘False
   EndProperty
   ForeColor       =        &H00FFFFFF&
   Height           =         285
   Left              =          360
   TabIndex        =           16
   Top              =          1200
   Width           =            1065
 End
 Begin VB.Label Label1
   Autosize      =    -1  'True
   Caption       =     "Item Name(Particular):"
   BeginProperty Font
      Name           =  “Times New Roman”
      Size            =       12
      Charset        =      0
      Weight         =      700
```

```
    Underline      =      0  ‘False
    Italic            =     -1  'True
    Strikethrough  =    0  ‘False
  EndProperty
  ForeColor       =       &H00FFFFFF&
  Height            =         285
  Left               =          360
  TabIndex        =            15
  Top                =          720
  Width             =          2415
End
Begin VB. Label IbILabels
  Autosize      =    -1  'True
  Caption        =     "Economic Order Quantity:"
  BeginProperty Font
    Name            =  “Times New Roman”
    Size              =       12
    Charset         =      0
    Weight          =      700
    Underline      =      0  ‘False
    Italic            =     -1  'True
    Strikethrough  =    0  ‘False
  EndProperty
  ForeColor       =       &H00FFFFFF&
  Height            =       285
  Index          =           4
  Left              =       360
  TabIndex      =       14
  Tag              =       "EconOrder Qnty:"
  Top               =       3120
  Width            =         2685
End
Begin VB.Label IbILabels
   Caption        =      "Reorder Level:"
   BeginProperty Font
     Name           =  “Times New Roman”
     Size           =     12
     Charset        =     0
     Weight         =     700
     Underline      =     0  ‘False
     Italic         =    -1  'True
     Strikethrough  =    0  ‘False
   EndProperty
   ForeColor      =      &H00FFFFFF&
   Height         =        255
   Index        =          3
   Left           =         360
   TabIndex      =          13
   Tag            =       "Reorder Level:"
   Top            =       2640
   Width          =         1815
End
Begin VB.Label IbILabels
   Caption      =     "Unit Selling Price:”
   BeginProperty Font
     Name           =  “Times New Roman”
     Size           =     12
     Charset        =     0
     Weight         =     700
     Underline      =     0  ‘False
     Italic         =    -1  'True
     Strikethrough  =    0  ‘False
   EndProperty
   ForeColor      =      &H00FFFFFF&
   Height         =        255
   Index        =          2
   Left           =         360
   TabIndex      =          12
   Tag              =        "UnitSellPrice:"
   Top               =          2160
   Width            =          2655
 End
 Begin VB.Label IbILabels
   Autosize      =    -1  'True
   Caption       =     "Quantity-On-Hand:"
   BeginProperty Font
     Name           =  “Times New Roman”
     Size            =       12
     Charset        =      0
     Weight         =      700
     Underline      =      0  ‘False
     Italic            =     -1  'True
     Strikethrough  =    0  ‘False
   EndProperty
   ForeColor      =       &H00FFFFFF&
   Height          =         285
   Index          =             1
   Left              =           360
   TabIndex      =           11
   Tag              =        " QntyOnHand:"
   Top               =        1680
   Width            =          1995
End
Begin VB.Label IbILabels
   Autosize      =    -1  'True
   Caption       =     "Part No:"
   BeginProperty Font
     Name           =  “Times New Roman”
     Size            =       12
     Charset        =      0
     Weight         =      700
     Underline      =      0  ‘False
```

```
      Italic            =     -1  'True
      Strikethrough  =    0  ‘False
    EndProperty
    ForeColor       =       &H00FFFFFF&
    Height           =         255
    Index         =           0
    Left              =          360
    TabIndex      =           7
    Tag              =        "PartNo:"
    Top                =         240
    Width            =          1815
  End
End
Attribute VB_Name  = “frmSTOCKS"
Attribute VB_GlobalNameSpace  = False
Attribute VB_Creatable  = False
Attribute VB_Predeclaredld  = True
Attribute VB_Exposed  = False
Option Explicit
Dim PartNo As String
Private  Sub cmdAdd_Click()
  On Error GoTo ErrorTrap
  PartNo   =  Trim(txtItemPartNo.Text)
  If Len(PartNo)>0 Then Call AddToListBox(PartNo)
  PartNo   =  ""
  datStocks.Recordset.AddNew
  Exit Sub
ErrorTrap:
  MsgBox  "This Action Produces An Error! . . Repeat!", vbOKOnly, "Database
Error Info"
  Screen.MousePointer  =   vbArrow
End Sub

Private Sub cmdClose_Click()
```

```
  Unload Me
End Sub

Private Sub cmdDelete_Click()
    'This may produce an error if you delete the last record or the only record in the recordset
    If datStocks.Recordset.RecordCount =   0 Then
        MsgBox   "There is no record in the table!"
        Exit  Sub
    End If
    If  MsgBox("Do You Want To Delete This Record!", vbYesNo, "Record Delete
  Dialog")  =  vbNo Then
        Exit Sub
    End If
    PartNo   =  Trim(txtItemPartNo.Text)
    Call DeleteFromListBox(PartNo)
    PartNo   =  ""
    datStocks.Recordset.Delete
    With datStocks.Recordset
          ‘.Delete
           .MoveNext
           If. EOF And. RecordCount >0  Then. MoveLast
    End With
End Sub

Private Sub cmdUpdate_Click()
    'datStocks.UpdateRecord
    ‘datStocks.Recordset.Bookmark  = datStocks.Recordset.lastModified
    ‘If Trim(UCase(PartNo))  <> Trim(UCase(txtItemPartNo.Text)) Then Call UpdateListBox
End Sub

Private  Sub AddToListBox(xStr)
    Dim Found As Boolean, i As Integer
    Found   =   False
    For i  =  0 To ListPartNo.ListCount  - 1
```

```
        If UCase(Trim(ListPartNo.List(i))) = UCase(Trim(xStr)) Then Found = True:
Exit For
    Next
    If Not Found Then List PartNo.AddItem xStr
End Sub
Private Sub Delete FromListBox(xStr)
    Dim Found As Boolean, i As integer
    Found = False
    For i = 0 To ListPartNo.ListCount - 1
        If UCase(Trim(ListPartNo.List(i))) = UCase(Trim(xStr)) Then
            Found = True: ListPartNo.RemoveItem i: Exit for
        End If
    Next
End Sub
Private Sub UpdateListBox()
    If Trim(UCase(PartNo)) <> Trim(UCase(txtItemPartNo.Text)) Then
        Call DeleteFromListBox(Trim(PartNo))
        Call AddToListBox(Trim(txtItemPartNo.Text))
    End If
End sub
Private Sub datStocks_Error(DataErr As Integer, Response As Integer)
    'This is where you would put error handling code
    'If you want to ignore errors, comment out the next line
    ‘If you want to trap them, add code here to handle them
    MsgBox "Data error event hit err:" & Error$(DataErr)
    Response = 0 ‘throw away the error
End Sub

Private Sub datStocks_Reposition()
    Screen. MousePointer = vbDefault
    On Error Resume Next
    ' This will display the current record position
    ' for dynasets and snapshots
    datStocks.Caption = "Record: " & (datStocks.Recordset.AbsolutePosition + 1)
```

```
   'for the table object you must set the index property when
   'the recordset gets created and use the following line
   'datStocks.Caption  =  'Record:  " & (datStocks.Recordset.RecordCount *
   (datStocks.Recordset.PercentPosition * 0.01)) + 1
   PartNo  =  txtItemPartNo.Text
End Sub

Private Sub datStocks_Validate(Action As Integer, Save As Integer)
   'This is where you put validation code
   'This event gets called when the following actions occur
   Select Case Action
      Case vbDataActionMoveFirst
      Case vbDataActionMovePrevious
      Case vbDataActionMoveNext
      Case vbDataActionMoveLast
      Case vbDataActionAddNew
      Case vbDataActionUpdate
      Case vbDataActionDelete
      Case vbDataActionFind
      Case vbDataActionBookmark
      Case vbDataActionClose
      Screen.MousePointer  =  vbDefault
   End Select
   Screen. MousePointer  =  vbHourglass
End Sub

Private  Sub Form_Activate()
   datStocks.ReadOnly  =  False
   datStocks.RecordsetType  =  0
   datStocks.Refresh
   datStocks.Recordset.AddNew
End Sub

Private  Sub Form_load()
```

```
  datStocks.ReadOnly  =  False
  datStocks.RecordsetType   =  0
  datStocks.Refresh
  With datStocks.Recordset
      .MoveFirst
      Do While Not .EOF
          If .EOF Then Exit Do
          ListPartNo.AddItem. Fields("PartNo”).Value
           .MoveNext
      Loop
  End With
  datStocks.Refresh
  datStocks.Recordset.AddNew
  datStocks.Recordset.Edit
End Sub

Private  Sub Form_Unload(Cancel As Integer)
  Screen.MousePointer  =  vbDefault
End Sub

Private  Sub txtItemPartNo_LostFocus()
   Dim Found As Boolean,  i%
   Found  =  False
   For  i%  = 0 To ListPartNo.ListCount  -  1
      If  UCase(Trim(ListPartNo.List(i%)))  =  UCase(Trim(txtItemPartNo.Text))
Then
          Found  = True:  Exit For
      End If
   Next
   If Not Found Then Exit Sub
   MsgBox  "PartNo  Already  Exists!",  vbOKOnly  +  vbInformation, "Stock Record Entry Error
 Dialog"
   txtItemPartNo.SetFocus
 End Sub
```

```
Private  Sub RefillListBox()
  With datStocks.Recordset
      .MoveFirst
      Do While Not .EOF
           ListPartNo.AddItem .Fields("PartNo").Value
           .MoveNext
      Loop
  End With
End Sub

Private  Sub EmptyListBox()
   ListPartNo.Clear
End Sub
```

```
'frmPurchase Form Module
VERSION 5.00
Object          =     "{FAEEE763-117E-101B-8933-08002B2F4F5A}#1.1#0";
"DBLIST32.OCX"
Begin VB. Form frmPURCHASE
  BorderStyle      =   3 'Fixed Dialog
  Caption          =  " Purchase Transaction Edit Form "
  ClientHeight     =   3705
  ClientLeft       =   1095
  ClientTop        =    1845
  ClientWidth      =    5940
  LinkTopic        =  " Form1"
  MaxButton        =   0  'False
  MinButton        =   0  'False
  ScaleHeight     =    3708
  ScaleWidth      =    5940
  ShowInTaskbar  =  0  'False
  Begin VB. TextBox txtPartNo
    DataField          =       "PARTNO"
    DataSource       =       "datPrimaryRS"
    Height             =        285
    Left               =       3120
    TabIndex           =         15
    Top                =        840
    Visible            =        0  'False
    Width              =         975
 End
 Begin VB. Data datStocks
    Connect           =   "Paradox 5.x;"
    DatabaseName  =     "C:\BlessedSuperstores"
    DefaultCursorType  =  0   'DefaultCursor
    DefaultType          =       2   'UseODBC
    Exclusive            =      0   'False
    Height               =       300
```

```
   Left                  =       4440
   Options               =        0
   ReadOnly             =        0   'False
   RecordsetType    =       0  'Table
   RecordSource     =       "Stocks"
   Top                   =        1080
   Visible               =        0  'False
   Width                 =       1215
End
Begin MSDBCtls.DBCombo DBComboPartNo
   Bindings          =       "frmPURCHASE.frx":0000
   DataField         =       "PARTNO"
   DataSource      =        "datPrimaryRS"
   Height             =        315
   Left              =          3120
   TabIndex       =            1
   Top                =          840
   Width             =        1215
   _ExtentX            =      2143
   _ExtentY            =       556
   _Version            =       327680
   ListField              =      "PartNo"
   Text                   =      "DBCombo 1"
End
Begin VB .PictureBox PicButtons
   Align              =       2  'Align Bottom
   Appearance      =       0  'Flat
   BorderStyle       =      0   'None
   ForeColor         =        &H80000008&
   Height                 =       540
   Left                   =        0
   ScaleHeight         =        540
   ScaleWidth           =      5940
   TabIndex            =          10
```

```
    Top                   =       2820
    Width                 =        5940
Begin VB. CommandButton cmdClose
    Caption        =      "&Close"
    BeginProperty Font
       Name            =  “Times New Roman”
       Size             =     12
       Charset         =      0
       Weight          =      700
       Underline       =      0  ‘False
        Italic             =      0  ‘False
        Strikethrough  =    0  ‘False
    EndProperty
    Height            =        420
    Left               =          4505
    TabIndex       =           14
    Top                =           0
    Width             =         975
 End
 Begin VB. CommandButton cmdBrowse
    Caption        =      "&Browse"
    BeginProperty Font
        Name            =  “Times New Roman”
        Size             =       11.25
        Charset         =      0
        Weight          =      700
        Underline       =      0  ‘False
        Italic             =      0  ‘False
        Strikethrough  =    0  ‘False
    EndProperty
    Height            =        420
    Left               =          3240
    TabIndex       =           13
    Top             =          0
```

```
  Width        =       975
End
Begin VB.CommandButton cmdDelete
  Caption      =    "&Delete"
  BeginProperty Font
     Name          =  “Times New Roman”
     Size          =     11.25
     Charset       =      0
     Weight        =      700
     Underline     =      0  ‘False
     Italic        =      0  ‘False
     Strikethrough =      0  ‘False
  EndProperty
  Height         =      420
  Left           =      1920
  TabIndex       =      12
  Top            =       0
  Width          =      975
End
Begin VB.CommandButton cmdAdd
  Caption      =    "&Add"
  BeginProperty Font
     Name          =  “Times New Roman”
     Size          =     12
     Charset       =     0
     Weight        =     700
     Underline     =      0  ‘False
     Italic        =      0  ‘False
     Strikethrough =      0  ‘False
  EndProperty
  Height         =      420
  Left           =      600
  TabIndex       =       11
  Top            =       0
  Width           =         975
 End
End
Begin VB.Data datPrimaryRS
   Align             =       2  'Align Bottom
   Caption           =   "  "
   Connect           =    "dBASE  IV;"
   DatabaseName   =      "C:\BlessedSuperstores"
   DefaultCursorType  =   0   'DefaultCursor
   DefaultType          =        2   'UseODBC
   Exclusive            =       0  ' False
   Height               =       345
   Left                 =          0
   Options               =          0
   ReadOnly            =           0   'False
   RecordsetType      =            1   'Dynaset
   RecordSource       =          "select * from [PURCHASE]"
   Top                 =           3360
  Width                =           5940
End
Begin VB.TextBox txtFields
   DataField   =      "AMOUNT"
   DataSource      =     "datPrimaryRS"
   Height               =         285
   Index               =          4
   Left                 =          3120
   TabIndex         =           4
   Top               =           2280
   Width              =           1455
End
Begin VB.TextBox txtFields
   DataField   =      "UNITCOST"
   DataSource      =     "datPrimaryRS"
   Height               =         285
   Index            =        3
   Left             =        3120
   TabIndex         =        3
   Top              =        1800
   Width            =        1335
 End
Begin VB.TextBox txtFields
   DataField   =     "QNTYPURCH"
   DataSource     =    "datPrimaryRS"
   Height           =        285
   Index            =        2
   Left             =        3120
   TabIndex         =        2
   Top              =        1320
   Width            =        1335
End
Begin VB.TextBox txtFields
   DataField        =     "TRADATE"
   DataSource       =    "datPrimaryRS"
   Height           =       285
   Index            =       0
   Left             =       3120
   MaxLength        =       10
   TabIndex         =        0
   Top              =        360
   Width            =       2295
End
Begin VB. Label lblLabels
   Caption       =     "Total Amount"
   BeginProperty Font
      Name           =  “Times New Roman”
      Size           =      12
      Charset        =     0
      Weight         =      700
      Underline     =     0  ‘False
      Italic        =    -1  'True
      Strikethrough =    0  ‘False
   EndProperty
   ForeColor        =    &H00FFFF00&
   Height         =       255
   Index          =        4
   Left           =       600
   TabIndex        =        9
   Top            =       2280
   Width          =       1815
 End
 Begin VB.Label lblLabels
   Caption       =     "Item Unit Cost"
   BeginProperty Font
      Name          = “Times New Roman”
      Size          =      12
      Charset       =     0
      Weight        =     700
      Underline     =     0  ‘False
      Italic        =    -1  'True
      Strikethrough =    0  ‘False
   EndProperty
   ForeColor      =      &H00FFFFFF&
   Height         =       255
   Index          =        3
   Left           =        600
   TabIndex        =        8
   Top            =       1800
   Width          =       1815
 End
 Begin VB.Label lblLabels
    Caption       =     "Quantity Purchased"
    BeginProperty Font
```

```
      Name          = “Times New Roman”
      Size          =     12
      Charset       =    0
      Weight        =    700
      Underline     =     0  ‘False
      Italic        =    -1  'True
      Strikethrough =    0  ‘False
   EndProperty
   ForeColor      =      &H00FFFF00&
   Height         =       255
   Index          =        2
   Left           =        600
   TabIndex       =          7
   Top            =       1320
   Width          =       2055
End
Begin VB.Label lblLabels
   Caption      =    "PartNo"
   BeginProperty Font
      Name          = “Times New Roman”
      Size          =     12
      Charset       =    0
      Weight        =    700
      Underline     =     0  ‘False
      Italic        =    -1  'True
      Strikethrough =    0  ‘False
   EndProperty
   ForeColor      =      &H00FFFF00&
   Height         =       255
   Index          =       1
   Left           =       600
   TabIndex       =          6
   Top            =       840
   Width          =       1815
```

```
End
Begin VB.Label lblLabels
   Caption       =     "Date:"
   BeginProperty Font
       Name           =  “Times New Roman”
       Size            =     12
       Charset        =     0
       Weight         =      700
       Underline      =      0  ‘False
       Italic            =     -1  'True
       Strikethrough  =    0  ‘False
   EndProperty
   ForeColor       =       &H00FFFF00&
   Height            =        255
   Index            =         0
   Left               =        600
   TabIndex        =          5
   Top               =        360
   Width            =        1815
  End
 End
Attribute VB_Name  = “frmPURCHASE"
Attribute VB_GlobalNameSpace  = False
Attribute VB_Creatable  = False
Attribute VB_Predeclaredld  = True
Attribute VB_Exposed  = False
Option Explicit

Private Sub cmdAdd_Click()

   If datPrimaryRS.Recordset.EditMode > 1 Then
          If Len(Trim(TxtFields(0).Text))  >  0 Then
              datStocks.Recordset.Index  =  "Stocks#PX"
              datStocks.Recordset.Seek  "=", DBComboPartNo.Text
```

```
            datStocks.Recordset.Edit
            datStocks.Recordset.Fields("QntyOnHand").Value
    datStocks.Recordset.Fields("QntyOnHand").Value _
             + Val(TxtFields(2).Text)
            datStocks.Recordset.Update
            datPrimaryRS.Recordset.Update
          End If
   End  If

   datPrimaryRS.Recordset.AddNew

 End Sub

 Private Sub cmdBrowse_Click()
  If  LCase(Trim(cmdBrowse.Caption))  =  "&browse"  Then
     cmdBrowse.Caption  =  "&Edit"
     datPrimaryRS.Visible  =  True
     txtPartNo.Visible   =  True
     DBComboPartNo.Visible   =   False
     cmdAdd.Enabled   =   False
     cmdDelete.Enabled   =    False
  Else If  LCase(Trim(cmdBrowse.Caption))  =  "&edit"  Then
     cmdBrowse.Caption  =  "&Browse"
     datPrimaryRS.Visible  =  False
     txtPartNo.Visible   =  False
     DBComboPartNo.Visible   =   True
     cmdAdd.Enabled   =   True
     cmdDelete.Enabled   =     True
     DBComboPartNo.Text  =txtPartNo.Text
  End If
End Sub

Private Sub cmdDelete_Click()
    On Error GoTo ErrorTrap
    If  MsgBox("Delete This Record? ", vbYesNo, "Delete Record Dialog")  =  vbYes
     Then
      With datPrimaryRS.Recordset
          .Delete
          .MoveNext
          If .EOF Then. MoveLast
      End With
      End If
      Exit Sub
    ErrorTrap:
       Screen.MousePointer  =  vbArrow
End Sub

'Private Sub cmdUpdate_Click()
       'datPrimaryRS.UpdateRecord
       'datPrimaryRS.Recordset.Bookmark  =  datPrimaryRS.Recordset.LastModified
End Sub

Private SubcmdClose_Click()
       Screen.MousePointer  = vbDefault
       Unload Me
End Sub

Private Sub datPrimaryRS_Error(DataErr As Integer, Response As Integer)
   'This is where you would put error handling code
   'If you want to ignore errors, comment out the next line
   'If you want to trap them, add code here to handle them
   MsgBox  "Data error event hit err:" & Error$(DataErr)
   Response  =   0 'Throw away the error
 End Sub

 Private Sub datPrimaryRS_Reposition()
      Screen.MousePointer = vbDefault
      On Error Resume Next
    'This will display the current  position for dynasets and snapshots
     datPrimaryRS.Caption     =     "Record:" &(datPrimaryRS.Recordset.AbsolutePosition +1)
     DBComboPartNo.Text  =  txtPartNo.Text
End Sub

Private Sub datPrimaryRS_Validate(Action As Integer, Save As Integer)
    'This is where you put validation  code
    'This event gets called when the following actions occur
    Select Case Action
      Case vbDataActionMoveFirst
      Case VbDataActionMovePrevious
      Case VbDataActionMoveNext
      Case VbDataActionMoveLast
      Case VbDataActionAddNew
      Case VbDataActionUpdate
      Case VbDataActionDelete
      Case VbDataActionFind
      Case VbDataActionBookmark
      Case VbDataActionClose
      Screen.MousePointer  =  vbDefault
    End Select
      Screen. MousePointer  = vbHourglass
 End Sub

 Private Sub DBComboPartNo_LostFocus()
      txtPartNo.Text   =  DBComboPartNo.Text
End Sub

Private Sub Form_Load()
      datPrimaryRS.Refresh
      datPrimaryRS.Recordset.AddNew
      datPrimaryRS.Visible    =   False
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
    Screen.MousePointer =  vbDefault
End Sub

Private Sub txtFields_GotFocus(Index As Integer)
    'If Index <> 4 Then Exit Sub
    TxtFields(4). Text  = Str(Val(TxtFields(2).Text) * Val(TxtFields(3).Text))
End Sub

Private Sub txtFields_LostFocus(Index As Integer)
   If Index  =  3 Then TxtFields(4).Text   =   Str(Val(TxtFields(2).Text) * Val (TxtFields(3).Text))
   'If Index <> 0 Then Exit Sub
   If IsDate(TxtFields(0).Text)  = False And Len(Trim(TxtFields(0).Text))   > 0 Then
      MsgBox  "Date Entered Is Invalid.  Re-enter...",  vbOKOnly, "Date Entry Dialog"
      TxtFields(0). SetFocus
   End If
   TxtFields(0).Text = Format(TxtFields(0).Text, "dd/mm/yyyy”)
   TxtFields(0).SelStart   = 0
   TxtFields(0).SelLength = Len(TxtFields(0).Text)
End Sub
```

```
‘frmPurchaseReport Form Module
VERSION 5.00
Object      =       “{0BA686C6-F7D3-101A-993E-0000C0EF6F5E}#1.0#0”;
 "THREED32.OCX"
Object      =         “{00025600-0000-0000-C000-000000000046}#4.6#0”;
 "CRYSTL32.OCX"
Begin VB.Form frmPurchaseReport
BorderStyle   =     1   'Fixed Single
  Caption        =   “Purchase Report Form"
  ClientHeight   =     4545
  ClientLeft      =      210
  ClientTop       =      1845
  ClientWidth     =    7230
  LinkTopic        =   " Form2"
  MaxButton       =    0 'False
  MinButton        =    0 'False
  ScaleHeight     =    4545
  ScaleWidth      =     7320
  Begin VB.TextBox txtStartDate
     BeginProperty Font
        Name         =    "FTG- Normal"
        Size           =        9.75
        Charset      =         0
        Weight       =         400
        Underline  =          0  ‘False
        Italic          =          0  ‘False
        Strikethrough   =    0  ‘False
     EndProperty
     Height              =       495
     Left                  =       2160
     TabIndex          =          0
     Top                   =       2400
     Width               =       1935
  End
```

```
Begin VB.TextBox txtEndDate
    BeginProperty Font
        Name        =      "FTG- Normal"
        Size        =       9.75
        Charset     =        0
        Weight      =       400
        Underline  =         0  ‘False
         Italic        =         0  ‘False
         Strikethrough   =    0  ‘False
    EndProperty
    Height              =       495
    Left                =       4920
    TabIndex            =          1
    Top                 =       2400
    Width               =        1935
End
Begin VB.Data datPurchRpt
    Caption             =   "Data1"
    Connect             =    "dBASE  IV;"
    DatabaseName   =     "C:\BlessedSuperstores"
    DefaultCursorType  =   0   'DefaultCursor
    DefaultType           =       2   'UseODBC
    Exclusive              =       0  ' False
    Height                 =       405
    Left                   =        600
    Options                 =         0
    ReadOnly              =         0   'False
    RecordsetType    =         0  'Table
    RecordSource    =        "PURCHRPT"
    Top                     =          3600
    Visible                =          0   'False
    Width                   =         1215
End
 Begin Threed. SSFrame SSFrame2
```

```
Height          =      1335
Left            =      240
TabIndex        =      4
Top             =      1800
Width           =      6855
_Version          =    65536
_ExtentX        =     12091
_ExtentY          =    2355
_ StockProps      =     14
Caption           =      "Report Period"
ForeColor         =       62580
BeginProperty Font    {0BE35203-8F91-11CE-9DE3-000AA004BB851}
   Name           =      “FTG-Normal”
   Size           =     11.25
   Charset        =     0
   Weight         =      700
   Underline      =      0  ‘False
   Italic         =      0  ‘False
   Strikethrough  =      0  ‘False
EndProperty
Font 3D           =       2
Begin Crystal.CrystalReport CrystalRptPurchase
  Bindings        =     "frmPurchaseReport.frx":0000
  Left            =       3960
  Top             =       240
  _ExtentX        =     741
  _ExtentY          =   741
  _Version          =    262150
  ReportFileName   =    "C:\BlessedSuperStores\purchase.rpt"
 WindowControlBox   =    -1  'True
 WindowMaxButton    =     -1  'True
 WindowMinButton    =    -1  'True
End
Begin VB.Label lbeEnterDate
```

```
   Caption      =    “EnterDate(s)”
   BeginPropertyFont
      Name       =    “FTG-Normal”
      Size        =     9.75
      Charset     =      0
      Weight     =      700
      Underline   =    0  ‘False
      Italic         =    0  ‘False
      Strikethrough =  0  ‘False
   EndProperty
   ForeColor         =    &H0000FFFF&
   Height              =    375
   Left                  =    240
   TabIndex          =     8
   Top                  =     720
   Width               =      1575
End
Begin VB.Label lbeStartDate
    Caption          =       "Start Date"
    BeginProperty Font
       Name           =      “Times New Roman”
       Size             =         9
       Charset        =        0
       Weight         =       700
       Underline      =     0  ‘False
       Italic             =     -1 'True
       Strikethrough  =     0  ‘False
    EndProperty
    ForeColor       =    &H00FFFF80&
    Height           =        495
    Left               =        2520
    TabIndex        =          7
    Top               =         360
    Width             =         855
```

```
End
Begin VB.Label lbeEndDate
   Caption         =    "End Date"
   BeginProperty Font
      Name            =    “FTG-Normal”
      Size            =       9
      Charset         =       0
      Weight          =      700
      Underline       =    0  ‘False
      Italic          =    -1 'True
      Strikethrough  =    0  ‘False
   EndProperty
   ForeColor       =   &H00FFFF80&
   Height          =      495
   Left            =      5400
   TabIndex        =       6
   Top             =      360
   Width           =      855
End
Begin VB.Label  lbeTo
   Caption         =    "To"
   BeginProperty Font
      Name            =    “FTG-Normal”
      Size            =      9.75
      Charset         =       0
      Weight          =      700
      Underline       =    0  ‘False
      Italic          =    1'  True
      Strikethrough  =    0  ‘False
   EndProperty
   ForeColor       =   &H00FFFF80&
   Height          =      255
   Left            =      4080
   TabIndex        =       6
   Top              =         720
   Width             =         375
End
Begin Threed. SSPanel SSPanel2
   Align            =        2  'Align Bottom
   Height            =          2895
   Left              =           0
   TabIndex          =          9
   Top              =         1650
  Width            =         7320
   _Version            =       65536
   _ExtentX          =        12912
   _ExtentY            =        5106
   _StockProps       =         15
   BeginProperty Font    {0BE35203-8F91-11CE-9DE3-000AA004BB851}
      Name            =       “FTG-Normal”
      Size             =      11.25
      Charset         =      0
      Weight          =       700
      Underline       =       0  ‘False
      Italic            =       0  ‘False
      Strikethrough  =      0  ‘False
   EndProperty
   BevelWidth       =         2
   BorderWidth       =        4
   BevelOuter       =        1
   BevelInner       =        2
   FloodColor         =     32768
   Begin VB.Data datPurchaseDate
      Caption        =     " Data 1 "
      Connect        =    "Paradox  5.x;"
      DatabaseName   =    "C:\BlessedSuperstores"
      DefaultCursorType  =     0   'DefaultCursor
      DefaultType          =        2   'UseODBC
      Exclusive          =      0  ' False
      Height             =      405
      Left               =        1200
      Options             =        0
      ReadOnly           =        0   'False
      RecordsetType   =          0   'Table
      RecordSource    =          "PURCHDATE"
      Top                =          1800
      Visible             =           0   'False
      Width               =          1215
   End
   Begin VB.Data datPurchase
      Caption           =   " Data 1 "
      Connect          =   "dBASE IV;"
      DatabaseName   =   "C:\BlessedSuperstores"
      DefaultCursorType  =   0   'DefaultCursor
      DefaultType          =       2   'UseODBC
      Exclusive          =      0  ' False
      Height             =      405
      Left               =        600
      Options             =        0
      ReadOnly           =        0   'False
      RecordsetType   =          0   'Table
      RecordSource    =        "PURCHASE"
      Top                =          2280
      Visible             =           0   'False
      Width               =          1215
   End
   Begin VB.Data datStocks
       Caption     =   " Data 1 "
       Connect     =   "Paradox 5.x;"
       DatabaseName   =     "C:\BlessedSuperstores"
       DefaultCursorType  =   0   'DefaultCursor
       DefaultType          =       2   'UseODBC
      Exclusive          =      0  'False
      Height             =       405
      Left               =        600
      Options             =        0
      ReadOnly          =        0   'False
      RecordsetType   =          0   'Table
      RecordSource    =        "STOCKS"
      Top                 =          1560
      Visible             =          0   'False
      Width                =         1215
   End
   Begin Threed. SSCommand SScmdClose
      Height           =           855
      Left               =          4680
      TabIndex        =            3
      Top                =         1680
      Width            =          1455
      _Version        =          65536
      _ExtentX        =         2566
      _ExtentY          =       1508
      _StockProps    =         78
      Caption            =       “&Close”
      ForeColor        =        16711680
      BeginProperty Font    {0BE35203-8F91-11CE-9DE3-000AA004BB851}
         Name           =      “FTG-Normal”
        Size             =      15.75
        Charset        =      0
        Weight          =      700
        Underline      =      0  ‘False
        Italic             =      0  ‘False
        Strikethrough  =      0  ‘False
      EndProperty
      BevelWidth       =        4
      Font3D            =        2
```

```
End
Begin Threed. SSCommand SScmdExecute
    Height          =         855
    Left            =         2400
    TabIndex        =          2
    Top             =        1680
    Width           =        1455
    _Version          =      65536
    _ExtentX        =        2566
    _ExtentY          =      1508
    _StockProps     =         78
    Caption           =        "&Execute"
    ForeColor        =          65280
    BeginProperty Font    {0BE35203-8F91-11CE-9DE3-000AA004BB851}
       Name           =     “FTG-Normal”
       Size           =     14.25
       Charset        =     0
       Weight         =     700
       Underline      =     0  ‘False
       Italic         =     0  ‘False
       Strikethrough  =     0  ‘False
    EndProperty
    BevelWidth       =       4
    Font3D           =       2
   End
End
Begin  Threed. SSFrame SSFrame1
    Height          =        1095
    Left            =          240
    TabIndex        =          10
    Top             =          240
    Width           =        6735
    _Version          =      65536
    _ExtentX        =        11880
    _ExtentY          =       1931
    _StockProps       =        14
    Caption           =      “Report Type”
    ForeColor         =       65280
    BeginProperty Font    {0BE35203-8F91-11CE-9DE3-000AA004BB851}
      Name          =     “FTG-Normal”
      Size          =     11.25
      Charset       =     0
      Weight        =     700
      Underline     =     0  ‘False
      Italic        =     0  ‘False
      Strikethrough =     0  ‘False
    EndProperty
    Font3D            =     2
    ShadowStyle       =     1
End
Begin  Threed.SSOption  SSOptDailyPurchases
  Height          =        495
  Left            =        360
  TabIndex        =        11
  Top             =        360
  Width           =        2655
  _Version        =        65536
  _ExtentX        =        4683
  _ExtentY        =        873
  StockProps      =         78
    Caption           =       "Daily Purchases"
    ForeColor         =       33023
    BeginProperty Font    {0BE35203-8F91-11CE-9DE3-000AA004BB851}
      Name          =     “FTG-Normal”
      Size          =     11.25
      Charset       =     0
      Weight        =     700
      Underline     =     0  ‘False
```

```
        Italic            =      0  ‘False
        Strikethrough  =     0  ‘False
    EndProperty
    Value              =        -1  ’True
    Font3D            =         2
End
Begin Threed.SSOption  SSOptPeriodicPurchases
    Height           =          495
    Left               =          3600
    TabIndex        =           12
    Top                =          360
    Width            =          2775
    _Version            =      65536
    _ExtentX         =          4895
    _ExtentY           =        873
    _StockProps      =        78
    Caption             =       "Periodic Purchases"
    ForeColor          =       33023
    BeginProperty Font    {0BE35203-8F91-11CE-9DE3-000AA004BB851}
        Name            =      “FTG-Normal”
        Size             =      11.25
        Charset         =      0
        Weight          =      700
        Underline       =      0  ‘False
        Italic             =      0  ‘False
        Strikethrough  =      0  ‘False
    EndProperty
    Font3D            =         2
  End
End
Begin Threed. SSPanel SSPane1
    Height           =          1575
    Index            =            0
    Left               =            0
    TabIndex       =        13
    Top            =         0
    Width          =       7335
    _Version          =     65536
    _ExtentX       =       12938
    _ExtentY          =     2778
    _ StockProps      =       15
    BeginProperty Font    {0BE35203-8F91-11CE-9DE3-000AA004BB851}
        Name           =     “FTG-Normal”
        Size           =     11.25
        Charset        =     0
        Weight         =     700
        Underline      =     0  ‘False
        Italic         =     0  ‘False
        Strikethrough  =     0  ‘False
    EndProperty
    BevelWidth        =       2
    BorderWidth        =      4
    BevelOuter      =       1
    BevelInner      =       2
    FloodColor         =     32768
    Font3D            =       2
    Autosize          =       3
Begin Threed. SSPanel SSPanel2
    Height         =       1335
    Index          =         1
    Left           =        120
    TabIndex       =         14
    Top            =         120
    Width          =        7095
    _Version          =     65536
    _ExtentX       =       12515
    _ExtentY          =      2355
    _StockProps      =       15
```

```
            BeginProperty Font    {0BE35203-8F91-11CE-9DE3-000AA004BB851}
               Name          =      “FTG-Normal”
               Size          =      11.25
               Charset       =      0
               Weight        =      700
               Underline     =      0  ‘False
               Italic        =      0  ‘False
               Strikethrough =      0  ‘False
            EndProperty
            BevelWidth       =        2
            BorderWidth      =        4
            BevelOuter      =       1
            BevelInner      =       2
            Autosize          =       3
         End
      End
 End
Attribute VB_Name  = “frmPurchaseReport"
Attribute VB_GlobalNameSpace  = False
Attribute VB_Creatable  = False
Attribute VB_Predeclaredld  = True
Attribute VB_Exposed  = False
Option Explicit

Private  Sub Form_Load()
   txtEndDate.Visible   =  False
   lbeTo.Visible   =   False
   lbeStartDate.Visible   =   False
   lbeEndDate.Visible   =    False
   lbeEnterDate.Caption  = 'Enter Date:"
End Sub

Private Sub SScmdClose_Click()
   Screen.MousePointer   =    vbDefault
   Unload.Me
End Sub

Private  Sub SScmdExecute_Click()

    Dim Message As String, PartNoStr  As String

    If Len(txtStartDate.Text) = 0 Or IsDate(txtStartDate.Text)  =  False Then
         If Len(txtStartDate.Text) =  0 Then
             MsgBox  "Date Not Entered!", vbOKOnly, " Error Information Box"
         Else
             MsgBox   "Date Entered Is Invalid!   Re-enter", vbOKOnly,  "Error Information Box"
         End If
         txtStartDate.SetFocus
         txtStartDate.SelStart   =   0
         txtStartDate.SelLength  =   Len(txtStartDate.Text)
         Exit Sub
    End If
         txtStartDate.Text  = Format(txtStartDate.Text,  "dd/mm/yyyy")
       If SSOptPeriodicPurchases.Value  =  False  Then  txt EndDate.Text   =  txtStartDate.Text
         If SSOptPeriodicPurchases.Value  =  True  Then
           If IsDate(txtEndDate.Text)  =   False  Or Len(txtEndDate.Text)   =   0   Then
             If Len(txtEndDate.Text)   =   0   Then
                MsgBox  "Date Not Entered!",  vbOKOnly, " Error  Information  Box"
             Else
                MsgBox   "Date Entered Is Invalid!   Re-enter", vbOKOnly,  "Error Information Box"
             End If
             txtEndDate.SetFocus
             txtEndDate.SelStart   =   0
             txtEndDate.SelLenght  =   Len(txtEndDate.Text)
             Exit Sub
        End If
        txtEndDate.Text = Format(txtEndDate.Text, “dd/mm/yyyy”)
        If CDate(txtStartDate.Text) >CDate(txtEndDate. Text) Then
```

```
                Message = “End Date:” & txtEndDate.Text & “ Cannot be less than StartDate:”
              &,txtStartDate.Text
                MsgBox Message, vbOKOnly, “Date Entry Error Information Box”
                 txtEndDate.SetFocus
                 txtEndDate.SelStart = 0
                 txtEndDate.SelLength = Len(txtEndDate.Text)
                 Exit Sub
         End If
End If
'Issue query on sales table
datPurchase.Refresh
datStocks.Refresh
datPurchRpt.Refresh
datStocks.Recordset.Index  =  'Stocks#PX"
datPurchRpt.Recordset.Index  =  'PurchRpNdx"
datPurchase.Recordset.Index  =  'DateNoNPX"
datPurchase.Recordset.Seek  ">="  CDate(txtStartDate.Text)
If datPurchase.Recordset.NoMatch  = True Then
    MsgBox  "No Record Meets the Query Criteria!", vbOKOnly,  "Table Query Info"
    Exit Sub
Else
     If datPurchase.Recordset.Fields("TraDate") <=  CDate(txtEndDate.Text)
Then
         'deletes all records in table
             With datPurchRpt.Recordset
                 Do While No.EOF
                      .Delete
                      .MoveNext
                  Loop
           End With
      Else
           MsgBox  "No Record Meets The Query Criteria!", vbOKOnly,  "Table Query Info"
           Exit Sub
      End If
```

```
 Do While Not datPurchase.Recordset.EOF
   PartNoStr  = datPurchase.Recordset.Fields("PartNo")
   datStocks.Recordset.Seek  "=", PartNoStr
   With datPurchRpt.Recordset
     .AddNew
     .Fields("TraDate")  =  datPurchase.Recordset.Fields("TraDate")
     .Fields("PartNo")  =  PartNoStr
     .Fields("ItemType")  =  datStocks.Recordset.Fields("ItemType")
     .Fields("ItemName")  =  datStocks.Recordset.Fields("ItemName")
     .Fields("QntyPurch")  =  datPurchase.Recordset.Fields(" QntyPurch ")
     .Fields("UnitCost")  =  datPurchase.Recordset.Fields(" UnitCost")
     .Fields("Amount")  =  datPurchase.Recordset.Fields(" Amount ")
     .Update
     .AddNew
   End  With
   datPurchase.Recordset.MoveNext
   If datPurchase.Recordset.EOF  =  True  Then Exit Do
   Message  =  datPurchase.Recordset.Fields("TraDate").Value
   If CDate(Message)  >  CDate(txtEndDate.Text)  Then
     'MsgBox   "Yes  This Date:    "   &  Message    &    "is graeter than: "      &      txtEndDate.Text,
 vbOKOnly
     Exit Do
  End If
Loop

datPurchaseDate.Refresh
If datPurchaseDate.Recordset.Count  =   0 Then
datPurchaseDate.Recordset.AddNew
Else datPurchaseDate.Recordset.Edit
datPurchaseDate.Recordset.Fields("StartDate").Value = txtStartDate.Text
datPurchaseDate.Recordset.Fields("EndDate").Value   =
llf(SSOptDailyPurchases.Value,  " "   “To"  &  txtEndDate.Text)
datPurchaseDate.Recordset.Update
datPurchase.Recordset.MoveFirst
```

```
     CrystalRptPurchase.PrintReport
   End If
End Sub

Private  Sub SSOptDailyPurchases_Click(Value As Integer)
   txtEndDate.Visible   =  False
   lbeTo.Visible   =   False
   lbeStartDate.Visible   =   False
   lbeEndDate.Visible    =    False
   lbeEnterDate.Caption  =  "Enter Date:"
End Sub

Private  Sub SSOptPeriodicPurchases_Click(Value As Integer)
   txtEndDate.Visible   =  True
   lbeTo.Visible          =    True
   lbeStartDate.Visible   =   True
   lbeEndDate.Visible    =    True
   lbeEnterDate.Caption  =  "Enter Dates:"
End Sub

Private Sub txtEndDate_LostFocus()
    txtEndDate.Text  =  Format(txtEndDate.Text, "dd/mm/yyyy")
End Sub

Private Sub txtStartDate_LostFocus()
    txtStartDate.Text  =  Format(txtStartDate.Text, "dd/mm/yyyy")
End Sub
```