

AN INTERACTIVE INVENTORY SYSTEM
BASED ON DBASE II

by

SUTEERA S. GRAHAM

B. S., Emporia State University, 1978

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1984

Approved by:



Major Professor

LD
2668
R4
1984
G72
C. 2

ALL202 620043

TABLE OF CONTENTS

<u>Chapter</u>	<u>page</u>
I. INTRODUCTION	1
Problem	1
Data Oriented Analysis	2
Alternative Propositions	3
II. FEASIBILITY STUDY	5
Scope of the Study	5
Scope of Master's Research	5
System Requirements Overview	6
Current System	6
Specific System Problems	11
Objectives for the Proposed System	11
System Requirements	12
Storage Requirement Estimates	12
Design Alternatives	13
Initial Costs Comparison	14
System Considerations	16
Custom versus Software Packages	16
On-Line versus Batch	16
Mass Storage Devices	18
System's Capacities Limitations	19
Selection Criteria of Hardware and Software	21
Implementation Plan	23
Status of System	24
III. SYSTEM DESIGN	26
System Specification	27
System Structure	29
Control Structure	29
Data Structure	39
IV. IMPLEMENTATION	46
System Files and Descriptions	47
Item Master File	47
Transaction File	49
History File	49
Customer File	52
Vendor File	52

V.	SUMMARY AND FUTURE WORK	67
----	-----------------------------------	----

	REFERENCES	69
--	----------------------	----

<u>Appendix</u>	page
-----------------	------

A.	USER'S GUIDE	70
----	------------------------	----

	Introduction to Inventory Sytem	70
--	---	----

	The Accounting Cycle	72
--	--------------------------------	----

	Introduction for the Operator	73
--	---	----

B.	SAMPLE SYSTEM REPORTS	89
----	---------------------------------	----

C.	SYSTEM SOURCE CODE	102
----	------------------------------	-----

LIST OF FIGURES

<u>Figure</u>	<u>page</u>
1. Entity Diagram of Inventory System	7
2. Inventory Supply	9
3. Pharmacy and Central Preparation	10
4. Comparison of Design Alternatives	15
5. Inventory Management System	28
6. Inventory Control Overview	31
7. Perform Inventory Control	32
8. Maintain Inventory Item File	33
9. Maintain Vendor File	34
10. Enter Transaction	35
11. Process End of Period	36
12. Back-up of Inventory Files	37
13. Produce Reports	38
14. Structure of Item File	41
15. Structure of History File	42
16. Structure of Transaction File	42
17. Structure of Vendor File	43
18. Structure of Customer File	43
19. Database Schema	44
20. Structure of Item Record	48
21. Structure of Transaction Record	49

23.	Structure of Customer Record	52
24.	Structure of Vendor Record	53
25.	System Calling Tree	54
26.	Main Inventory Menu	55
27.	Master File Maintenance	55
28.	Vendor File Maintenance	59
29.	Transaction Menu	59
30.	End of Period Menu	62
31.	Report Menu	64
32.	File Backup Menu	65

Chapter I

INTRODUCTION

1.1 PROBLEM

The state of the store room at the Veterinary Hospital, Kansas State University was such that the investigation of computerizing the inventory control system was needed to provide accurate and timely management information.

Proper computerization of business systems involves three major activities which must be carried out before a design decision may be made. These three activities are fact finding, analysis of the business requirements and formalizing the analysis in the form of a report (4). The fact-finding activity required analyzing and documenting the existing manual procedures for stock control and maintenance. Detailed research was carried out to document the functional parts of the inventory system, to determine what functions may be automated and to determine the feasibility of such automation. The feasibility study allowed decisions to be made as to what could reasonably be required of an automated inventory system.

The next phase of system development involves preparation of a precise specification of the problem or functional description (4). The requirements derived from study of the

feasibility document allowed the system specification and design of the system to be developed and documented. This resulting design was analyzed to determine what parts of the system could be computerized. Improvements in the design of the existing system were made in order to improve efficiency, eliminate redundancy, increase accountability and otherwise provide for the tracking of the movement of inventory items.

1.2 DATA ORIENTED ANALYSIS

Analysis of the manual system of inventory management was based on the flow of information within the system. The use of the structure of the data to determine the system design expounded in several popular design methods (6, 13). Every program has an internal data base to maintain. This data base is a common characteristic and is the core of the function of all programs (7). Such data oriented design methodologies may have their emphasis on one of three areas. Input oriented methods generate the program data structure starting with the inputs and resulting with the required outputs. A characteristic of input oriented data driven design methods is that a small change in the input data definition may yield a large modification to the resulting program structure (6). The second type of data driven design is transformation oriented. The characteristic of this type of design is the efficiency found in commonality of general

modules in the resulting system (13). The third type of design is output oriented. In this method system outputs are defined and dissected to define the transformations and inputs required (9, 13). The advantage found with this method is that the inputs and transformations are the absolute minimum required (1). Input oriented design may result in a design encompassing inputs which turn out not to be absolutely necessary and superfluous transformations on those unneeded inputs.

The output oriented Warnier methodology was chosen for the design and documentation of the inventory control system both because of this obvious advantage and that sophisticated tools for the production and manipulations of Warnier diagrams exist. The automated Warnier production system produces the source code once the design is refined to the smallest detail.

1.3 ALTERNATIVE PROPOSITIONS

Alternative plans for computerization were produced for consideration:

- Continue with current system but improve the management procedures.
- Develop the system to run on the main-frame computer.
- Purchase the micro-computer (stand alone) system and software.

**THIS BOOK CONTAINS
NUMEROUS PAGE
NUMBERS THAT ARE
ILLEGIBLE**

**THIS IS AS RECEIVED
FROM THE
CUSTOMER**

- Purchase the micro-computer and software system and interface with the main-frame computer.

These alternatives were examined for feasibility, and, in conjunction with the client, a final system plan was developed for implementation. The development of the resulting system is presented in this paper.

- Chapter 2 presents the feasibility study as was prepared for the client, a discussion of the alternatives, and a description of the final proposed system.
- Chapter 3 contains the detailed system design. The system program and data structures are presented in Warnier Diagrams and system specifications are presented in tabular form.
- Chapter 4 presents the implementation of the system. The implementation language choice is justified. A description of the user menus and of the modular implementation of the system is included. This chapter concludes with a description of the implemented system, an evaluation of the completed product, and notes for future implementations of similar projects.
- Chapter 5 contains conclusions and a summary of a project.
- The appendix contains source code of the system and the user's guide.

Chapter II

FEASIBILITY STUDY

The objective of the study is to determine the requirements for computerizing the business functions of the Department of Surgery and Medicine, College of Veterinary Medicine at Kansas State University.

2.1 SCOPE OF THE STUDY

The study covers the major portion of the administrative operations of accounting and inventory management of the department of Veterinary Medicines. The accounting functions examined are billing, accounts receivable, cash receipts, and the periodic reports produced in support of these functions. The inventory management functions included in this study are inventory control and inventory forecast.

2.2 SCOPE OF MASTER'S RESEARCH

- Feasibility Study
- Design of Inventory System
- Implementation of Inventory System in dBASE II

2.3 SYSTEM REQUIREMENTS OVERVIEW

2.3.1 Current System

Each area operates independently of each other. Most areas use manual operations assisted by an accounting machine. IBM Displaywriter word processing stations are currently being used in some areas. See figure 1 for an entity diagram of the current system. An entity diagram is the result of entity analysis performed on the existing system. From fact-finding efforts a list of entities is produced. Entities are items, either tangible or intangible, which are transmitted or acted upon as part of the inventory control process (8). The entity diagram may be thought of as an alternative way of expressing the functional architecture which expresses the result of both context analysis and boundary definition (11). The bubbles represent objects and the arcs represent operations on these objects. The diagram is then a description of the data flow of the system, necessary information for the system design process (9).

Actual inventories of supplies and the pharmacy are conducted annually. There are presently no history records of supplies used. Stock status is only estimated and reordering is done when the supply of a requested item is noticeably low or exhausted. The following is the description of supply inventory process as shown in figure 2

1. When the supply is provided to dispensary areas, an inventory clerk updates his inventory cards file. If

**THIS BOOK
CONTAINS
NUMEROUS PAGES
WITH DIAGRAMS
THAT ARE CROOKED
COMPARED TO THE
REST OF THE
INFORMATION ON
THE PAGE.**

**THIS IS AS
RECEIVED FROM
CUSTOMER.**

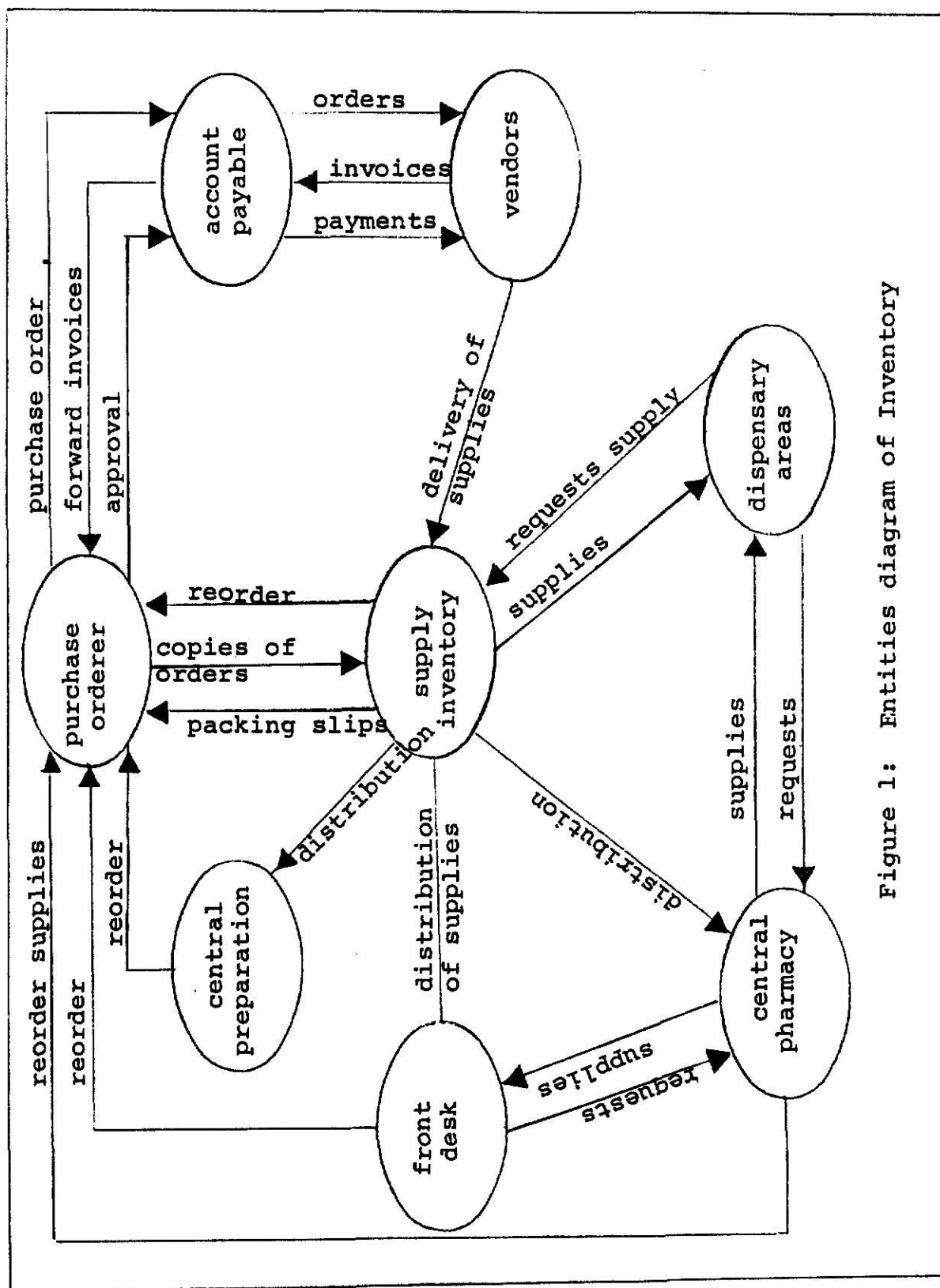


Figure 1: Entities diagram of Inventory

that supply item is out of stock or below the low limit, he writes down the description of the item and quantity to order and forwards the order to his supervisor.

2. The supervisor selects the vendor, prepares the purchase order and sends out his order. The copy of purchase order is sent back to the inventory clerk for comparison with the packing slip when the supplies arrive. The inventory clerk updates his inventory cards file again when he receives the supplies and distributes them to the specific storage rooms which are indicated on the copy of purchase order.

The descriptions of pharmacy inventory and central preparation procedures are much the same as supply inventory. See figure 3 for more detail.

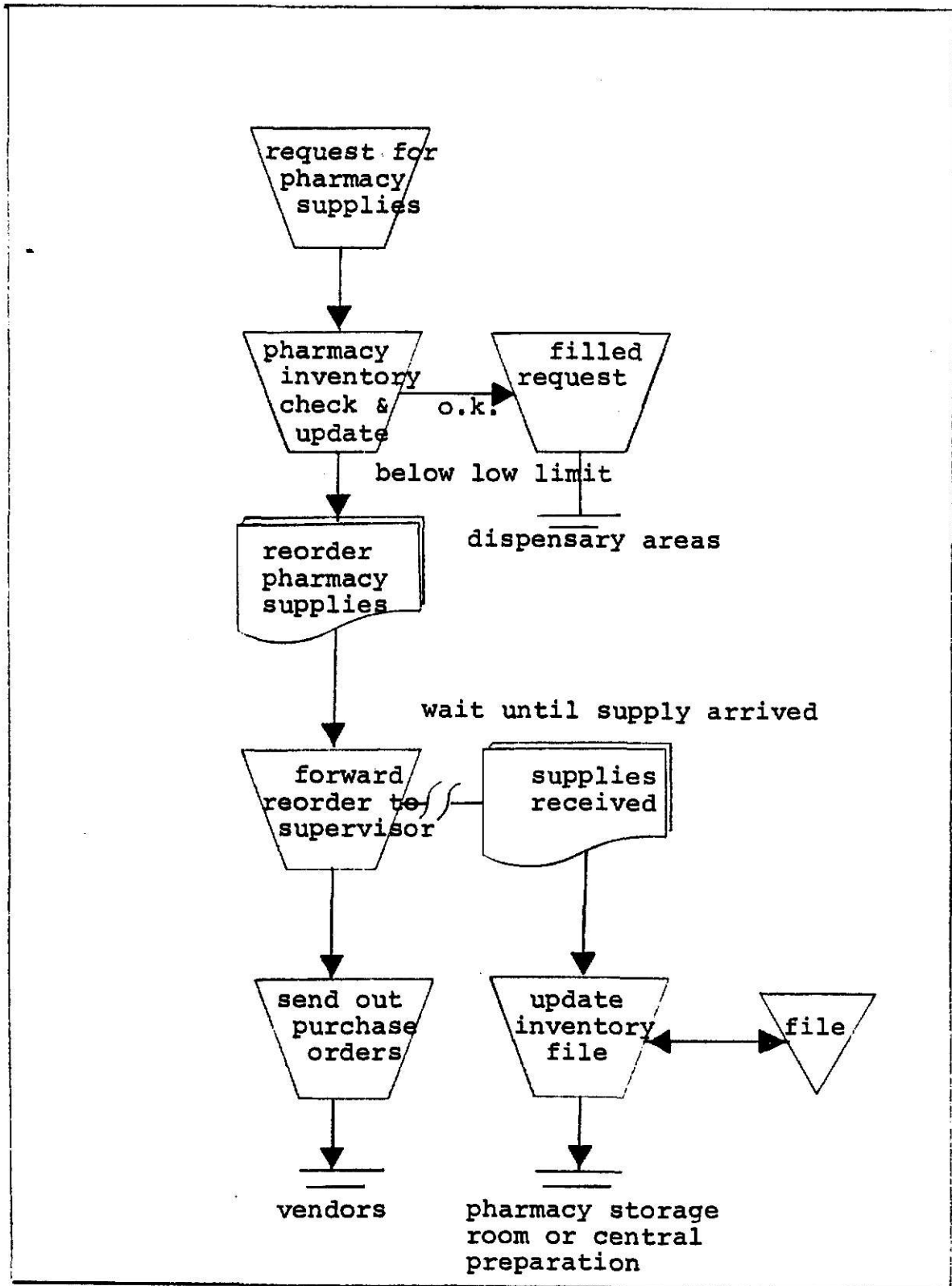


Figure 2: Inventory Supply

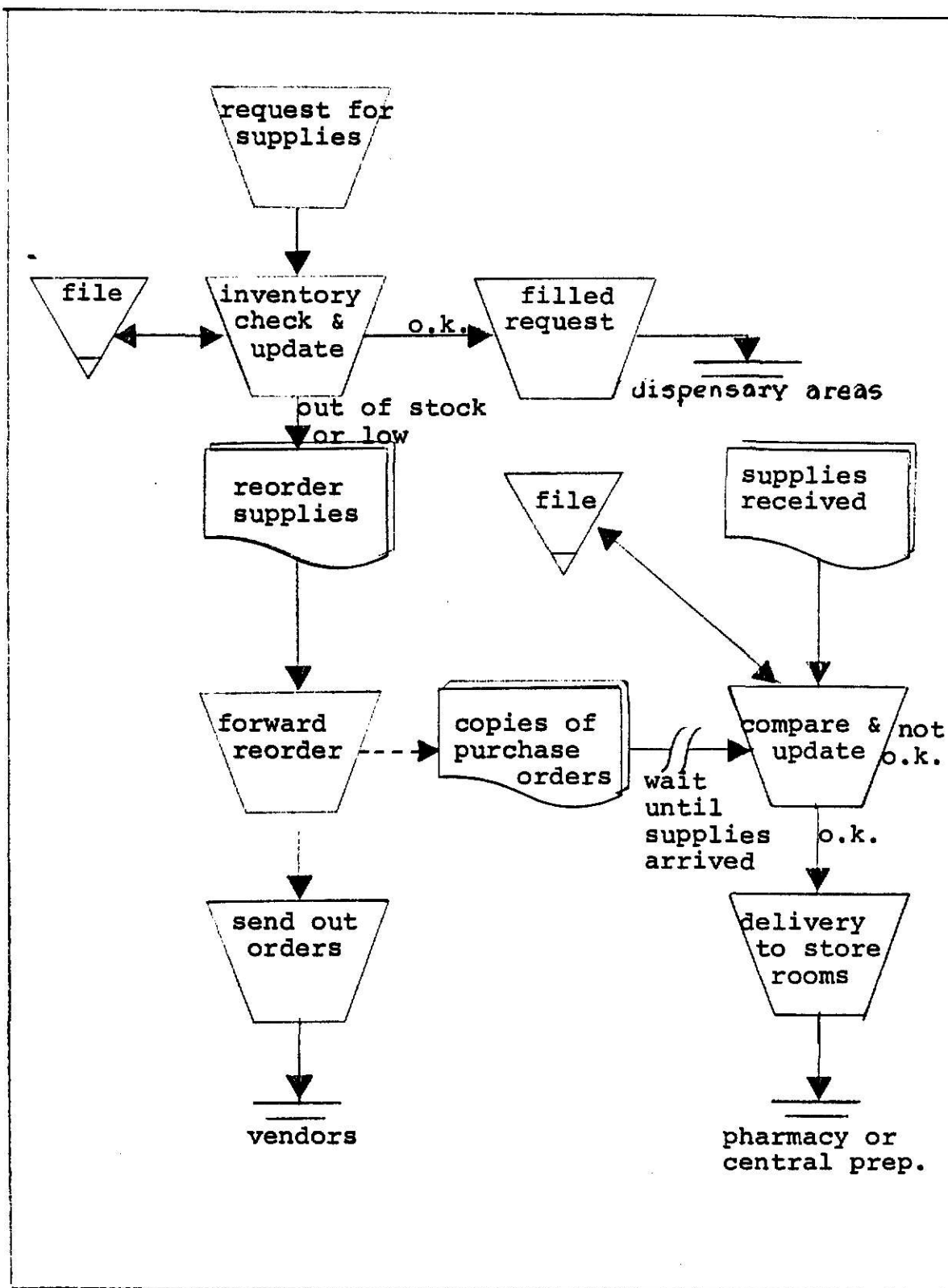


Figure 3: Pharmacy and Central Preparation

2.3.2 Specific System Problems

1. Improvements in the manual system are needed to insure the success of the computerized system. For example, insuring that all material movements in and out of supply are properly recorded.
2. There are occasionally no records of movement of supplies which have been taken from storage rooms. This causes inaccuracy in the inventory file.
3. There is no set reorder level for some inventory items. These must be established for computerized system to work.

2.3.3 Objectives for the Proposed System

1. Improved stock status report with accurate numbers of stock on hand and on order, the safety stock level, the historical uses, the price and cost information for each item.
2. Reduction of redundancies in inventory items among the storage rooms.
3. Analysis of distribution of inventory items to each area.
4. Freeing of key people from clerical operations.

2.3.4 System Requirements

The college of Veterinary Medicine plans to install a system that will provide the following:

1. Support at least 2,000 items of inventory
2. Allow daily analysis of current stock status report including use/month, use/year, expiration date, cost/unit, sale price/unit, vendor name and storage location.
3. Have the capability to automatically update inventory information from transaction input.
4. Have a capability to keep the history record of item movement from storage locations to dispensing areas.
5. Have compatibility to current system.

2.3.5 Storage Requirement Estimates

It was determined that three "master files" would be necessary to hold information on the primary entities involved in the system: the item file, the customer file and the vendor file. Record lengths for each of these files was estimated. In addition it was estimated that a transaction file of approximately 50 bytes and a history file of 926 bytes would be used to track inventory item movement. The inventory item record would need to contain information on description and price detail, reorder detail, current item activity detail and vendor. The customer file would contain information on customer code and customer name. The vendor file

would contain vendor code and vendor information. The history file would contain information on item code, customer code and historical detail.

Master File Data:

Inventory Master File	2,000 x 200 bytes =	400K bytes
Vendor File	250 x 137 bytes =	29K bytes
Customer File	30 x 36 bytes =	1K bytes

Transaction Data per year

Transaction File	12 x 500 x 50 bytes =	300K bytes
------------------	-----------------------	------------

History Data per year

History File	2,000 x 926 bytes =	1.9M bytes
--------------	---------------------	------------

Total Storage Required = 2.5M bytes

The system will require an approximate 2.5 million bytes of disk storage for one year.

2.4 DESIGN ALTERNATIVES

After studying the present system, there were six alternatives to be considered:

1. Continue with current system.
2. Purchase a software package to run on the NAS system in the Computing Center.
3. Develop a custom software system to run on the NAS system.
4. Purchase a micro-computer system and software (stand-alone) to perform data recording, editing, and reporting.

5. Purchase a micro-computer system and software packages to do processing with local storages and interface with the NAS system for back-up and print reports.
(To avoid the cost of high speed printer)
6. Purchase a micro-computer system and software package to do processing and interface with the NAS system for storages, back-up and print reports. (To avoid the high cost of hard disk and the cost of high speed printer)

2.4.1 Initial Costs Comparison

Although the user has not yet selected specific vendors, the following figure, figure 4, summarizes the approximate costs involved with the design alternatives.

ALTERNATIVES AND FACTORS	2 NAS AND SOFTWARE PACKAGE	3 NAS AND CUSTOM SOFTWARE	4 MICRO (STAND-ALONE) AND SOFTWARE PACKAGE	5 MICRO & SOFTWARE PACKAGE WITH NAS	6 MICROE SOFTWARE PACKAGE WITH NAS
SOFTWARE	\$5,000 (Info Package)	\$50,000 (CC)	\$1,300 (Peachtree) \$3,000 (Utilities)	\$1,300 (Peachtree) \$1,000 (Interface to CMS CC) \$3,000 (Utilities)	\$1,300 (Peachtree) \$2,000 (MICRO and Summaries) \$800 (UNIX)
TERMINALS	\$2,000x2 = \$4,000 (COURIER) plus \$5,000 Controller	\$2,000x2 = \$4,000 (COURIER) plus \$5,000 Controller	\$800 (Total of two)	\$800 (Total of two)	\$800 (Total of two)
PRINTER	-	-	\$6,000 (300 LPM)	\$700 (EPSON 80 CPS)	\$700 (EPSON 80 CPS)
MICROCOMPUTER	-	-	\$7,000 (PSI 8000) (up to 4 expansion terminals)	\$5,000 (DIGITAL) (PROFESSIONAL 350)	\$4,000 (IBM PC) (includes 2 320K-Byte diskette drives)
DISK STORAGE and OPERATING COST	\$600/yr Disk \$4,000/yr Reports \$10,000/yr Editing and Update	\$600/yr Disk \$2,500/yr Reports \$4,000/yr CMS Edit and Update	\$5,000-up each for hard disk (NO provision for Back up of Files except diskette)	\$5,000-up each for hard disk \$1,000/yr for Reports & Back-up	\$300/yr disk \$2,500/yr Reports \$1,000/yr CMS
HARDWARE MAINTENANCE	\$1,350/yr (15% of 9,000)	\$1,350/yr (15% of 9,000)	\$2,820/yr (15% of 18,800)	\$1,725/yr (15% of 11,500)	\$825/yr (15% of 5,500)
ADDITIONAL PERSONNEL	0	500 Hrs/yr \$20/hr = \$10,000 (PROGRAMMING)	\$19,100/yr (Programmer II)	500 Hrs/yr \$20/hr = \$10,000 (Programming)	500 Hrs/yr \$20/yr = \$10,000 (Programming)
ONE-TIME COST	\$14,000	\$59,000	\$23,100	\$16,800	\$9,600
CONTINUING COST PER YEAR	\$15,950	\$18,450	\$21,920	\$12,725	\$14,625
TOTAL COST (FIRST YEAR)	\$29,950-37,500	\$77,450-96,812	\$45,020-56,300	\$29,525-36,900	\$24,225-30,300

2.4.2 System Considerations

There are several systems-related considerations that must be included in the overall design.

Custom versus Software Packages

The decision as to whether custom software will be written or off-the-shelf software purchased is one of economics, not convenience. Aside from the potentially large difference in price, the challenge of implementation is equally great. In the case of custom software, the major problems will be designing, programming, and debugging the programs. There will be some problems to overcome before smooth operation is achieved since these programs are created and tested outside the business environment.

If software packages are selected, the problem will be one of adapting the business to constraints imposed by the purchased software. It is not likely that the day-to-day business operating policies will fit as well with software packages as they would on custom software.

In either case, the users should expect, anticipate, and plan on an extended period of disruption of the office during conversion to the automated system. Careful and detailed planning can do much to minimize this disruption. (1)

On-Line versus Batch

In a batch system, a set (perhaps a day's worth) of transactions are accumulated and processed as a batch to update the files of the system and produce reports. In an on-line system, each transaction is processed as it occurs, so the files are kept continuously updated. Even in an on-line system, reports are often obtained in "batch" mode at daily or weekly intervals.

The decision between use of on-line or batch system components follow from the costs and benefits in each case.

The batch-control technique has a relatively low cost; while the on-line approach, however, often has high cost of development and operation. On-line systems offers important features not found in batch system. For example, inventory can be decremented as orders are processed. Thus, the current inventory position is maintained perpetually, instead of being accurate for only a brief time after a batch is run. Not all business operations find advantage in on-line operation. Business procedures which involve collecting information distributed offices or stations and delivering them to a centralized filing point lose two of the advantages of online operation. First, since transactions which take place are not entered into the system at the time they occurs, real time inquiring of the inventory status is not available. Second, the online entry of these collected transactions will seem frustratingly slow since a large number of them would need to be entered in the system at a

time. This type of operation could be implemented sufficiently in batch mode. There is always a place for both methods of operation in every system. Consideration should be made during the system design stage as to whether the cost or the benefit is the most important factor in each instance. (4)

If the NAS is used, the cost of many hours of connection to the machine will virtually preclude an on-line system. If a microcomputer of sufficient capacity is purchased, on-line operation seems the natural choice, since the machine might as well be busy as idle.

Mass Storage Devices

Serial-Devices. The cassette recorder or tape is the simplest and least expensive of all mass storage devices. The cassette clearly has a place in data storage for a batch processing system but as the primary mass storage device in a business system it is inadequate. In microcomputer system, where the user might be accessing an item file with thousands of records, the tape approach could require hours to access needed items of information.

Random-Access Devices. The data are stored such that potentially they can be accessed randomly. That is, any piece of information is available at any instant of time in an on-line processing system. There are two types of random access memory.

1. Floppy disks are also called flexible disks or diskettes and are an outstanding storage medium from a cost-benefit point of view.
2. Hard disks provide greater volumes and shorter access times than floppy disks. Their price, beginning at about \$2,000 , appears high, but on a cost-per-byte basis they are a bargain if that much capacity is needed and rapid random access to support an on-line is required.

How many disk drives are needed in a system? Generally, at least two drives are essential. The reason for two as a minimum is the needed capacity and the ability to back up files. Back-up could be on a floppy disk or cassette, but as file size increases it is probable that a second hard disk will be required to back up the first. On the other hand, the backup files are seldom used and always read sequentially when used, so the limitations of primary storage device do not apply. In term of costs and benefits, the addition of a cassette or floppy disk as a backup device is well worth considering. (5)

System's Capacities Limitations

1. Disk Limitations on Microcomputer Systems. If a microcomputer is purchased for the bottom dollar, it seems likely that the total data storage requirement for the overall system will exceed the capacity of

the diskette drives that are attached to the CPU (central processing Unit). The operation of different functional systems would require diskette changes as the different subsystems are run which would preclude simultaneous on-line operation of all subsystems. Planning which files go on which diskettes to provide room for growth and balance between the storage disks would be a major design concern. Ad hoc files are often created for maintenance or report generation and even for that brief time may become sizable and require strategic planning. Not to be forgotten are the program storage decisions. Programs require storage space and must be available when needed. We caution against selection of a system too near the minimum required.

2. Memory Limitations on Microcomputer Systems. Today, large microcomputers can have many thousands of bytes of memory and millions more that behave almost as if they were memory (called virtual memory, even though the actual storage occurs on disk). In microcomputers there are still some limitations that need to be considered. Of the 64K bytes of memory available in a large microcomputer, 4K might be needed for the operating system and 12K more for the language. This leaves 48K bytes for programs and data. If the programs are large, the space for data and computations may be too small to achieve the objective. (5)

Good languages allow one program to call another from the disk, load that program into memory, and begin to run it. Thus, programs can be broken into modules that can be removed from storage as needed. This technique can overcome memory limitations but requires careful design and strategic placement of modules on diskettes so that the proper module is where it should be when needed.

However, microcomputers can be purchased in various memory sizes, with several peripheral devices and with many hardware and software features. Selecting the correct equipment with the desired amount of memory and with the required peripherals and features can be accomplished with a thorough systems design and a study of several microcomputer systems.

2.4.3 Selection Criteria of Hardware and Software

Purchased systems must be evaluated in order to determine the applicability, advantages and disadvantages of each. This requires the development of criteria by which the packages may be judged (1). The selection criteria focus on four broad considerations:

1. **Compatibility.** It is extremely important to have compatibility demonstrated by dealer if products of more than one vendor are to be mixed - or if what appears to be two product lines of one vendor are to be

mixed. This demonstration must be a reasonable test. If the components can not communicate, they are unacceptable because they will not work together. Further, if the components cannot satisfactorily communicate with the user (too slow, too small, etc.) they are unacceptable. (4) It is important to evaluate the performance of a software package on the same hardware as is proposed for the system. Packages which are adaptable to several operating systems may have significantly different performance characteristics in each of the environments (1).

2. Documentation. All hardware and software must be fully documented. There must be manuals that explain how to use each element. Compare the documentation from each vendor and learn how often they update the product and the documentation. (1)
3. Service and Support. All systems seem to fail occasionally, even those designed for continuous operation. Therefore, service is extremely important. The proposed vendor should provide written details about how service is performed: where it is done, in how timely a fashion, and who will do the work. The buyer needs to be completely assured that service will be there in two hours, in two months, a year from now and beyond before resources are committed. (1)

4. Vendor Stability and Reliability. The vendor should be considered carefully. Determine as much as possible about the company's size, track record, growth and so on. Any unstable environment may pose many problems. For example, if the company goes out of business , equipment owners are left with only promises: promises of unfulfilled software expansion, support, and development, of hardware enhancements, and so on - all of which probably will remain unavailable and unfulfilled. As a result, a new system is needed to continue with the project. First-rate vendors encourage user groups and foster communications. They provide enhancement when improvements can be made. They continue to support the line of product with new options, accessories, and power. A high quality vendor is a great asset. (1)

2.4.4 Implementation Plan

1. System Review and Selection.
2. System Design. When the final selection of system is made, the process of detailed design of output reports, processing files, and input data can be started.
3. System Programming and Documentation.
4. System Installation and Testing.
5. Staff Training.

6. Performance Review. To ensure that each of the areas is being improved, a performance review should be held on a regular basis with the development team.

2.4.5 Status of System

After reviewing the alternatives and searching for software, the department of Surgery and Medicine has decided to acquire a 16 bit word microcomputer system and custom designed and implemented software for their inventory system. The main-frame solution was ruled out due to the high cost of connect time required for a truly interactive online system and because complexity of the operating environment (IBM's VM/SP and CMS packages) would be too complex for clerical and support-level staff to learn. Extensive investigation of the software available on the market revealed no packages which were suitable or tailorable to meet the requirements and still be simple enough for operation by the intended user. The resulting configurations are given below.

1. Hardware Configuration. The inventory system will be running on Zenith Z-100 with a 10 megabytes hard disk, a floppy disk drive and a dot matrix printer with 200 characters per second (CPS). The operating system is Z-DOS (a version of MS-DOS).
2. Software Configuration. The custom inventory control system will be written in dBASE II, a relational database management system that allows easy manipula-

tion of small and medium sized databases. The program dBASE II has a query/report feature able to access the inventory information being maintained by the inventory system. As long as user-developed programs read information and do not modify it, there is no chance that such activity can interfere with the ongoing use of the system. Users are cautioned against making manual alternations in the contents of the system files if full dBASE version is used. The inventory system may be used in either a full dBASE version or in a dBASE "runtime" version. The dBASE II Runtime is the application development tool that allows dBASE II programmers to market software to user who do not have the full dBASE II system.

Chapter III

SYSTEM DESIGN

The design of a microcomputer software system requires a different viewpoint from the design of a system to run on a mainframe computer. On a microcomputer, the size of files, the record length and the number of fields in a record is limited and the speed of access to individual records of a file is slow, compared to mainframe computers. One difference with microcomputers is the capacity for "user-friendliness", i.e., the ability to provide menus which guide the user at every step of the way and makes it possible for a novice to become immediately effective at working with the computer program, given a familiarity with the application area. Another difference is that the availability of microcomputer services makes it customary to keep information constantly updated. This differs from the practice in batch systems on a mainframe where the custom is to make runs which update information only at intervals. The fact that current data on item availability is constantly maintained in the system means that managers can reference the system for current information at any time, without waiting for end of period runs to get up-to-date information.

3.1 SYSTEM SPECIFICATION

The inventory management system must produce a number of reports which can be used for:

1. auditing inventory management activity
2. statistical review of inventory
3. management review of inventory movement
4. determination of inventory value
5. inventory items analysis

The inventory control system must be able to manipulate the data in the inventory files in the following ways:

1. Add/Delete/Modify information on inventory items
2. Add/Delete/Modify information on vendors.
3. Track inventory items used by each customer.
4. Track inventory items supplied by each vendor.
5. Provide information on item reorder levels.
6. Produce reports as required by the client.

The following figure 2 present the system report functions in Warnier diagram (13). See appendix for sample output formats of the system.

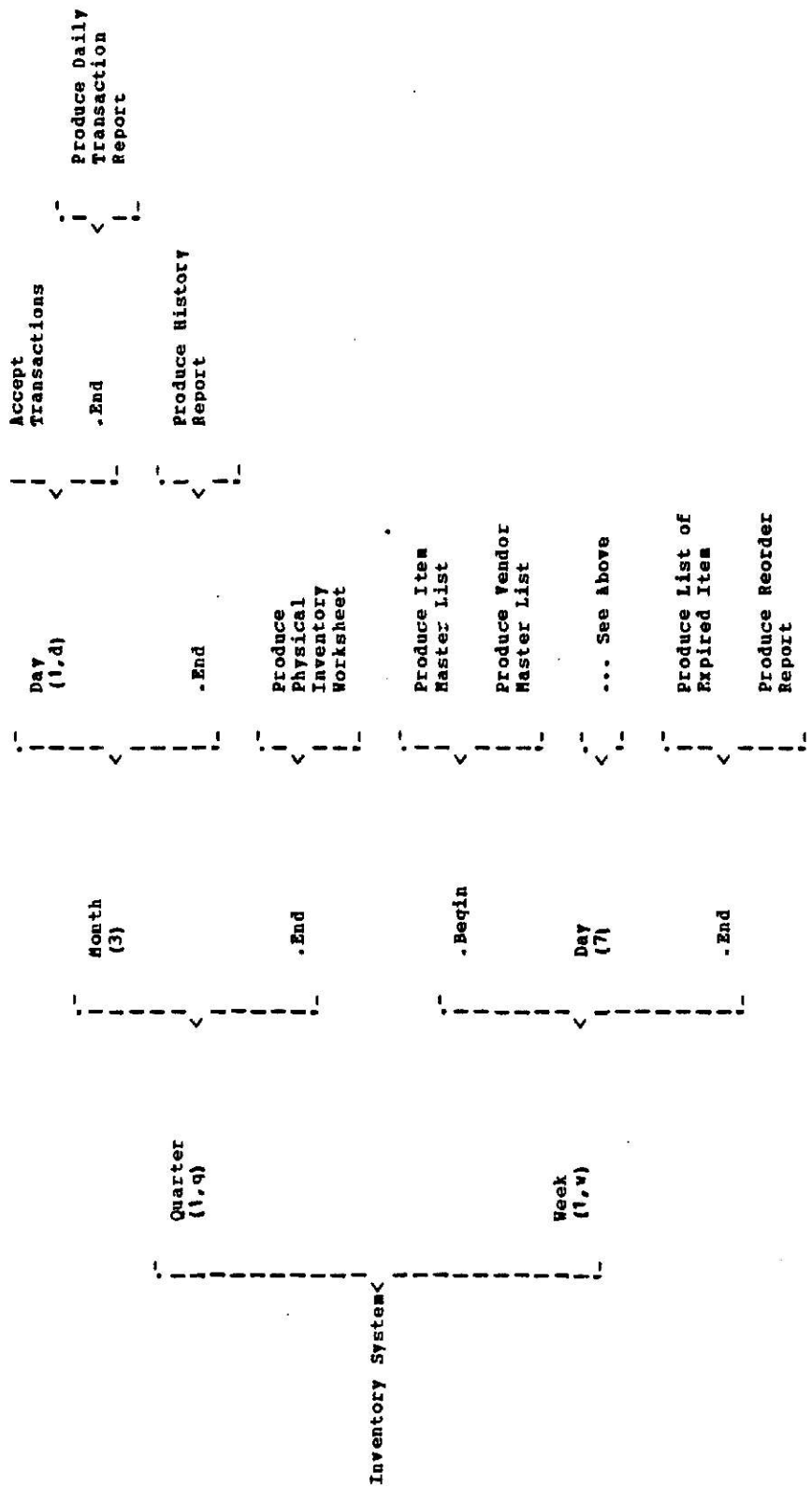


Figure 5: Inventory Management System

3.2 SYSTEM STRUCTURE

The system structure is made of two principal components: the program structure (control structure) and the file structure (data structure). The control structure defines the order in which operations are performed on the program database. The data structure defines how the data elements are organized and how relationships between data elements are shown in order to represent information about stockroom inventory. (6, 13)

The notation used during system design to represent both the control and data structure of the system is the Warnier diagram. The Warnier diagram is a method for representing an information hierarchy and depicts information as a treelike data structure (10). The Warnier diagrams shown in this report have been produced by a program written at the Kansas State University Computing Center. (3)

3.2.1 Control Structure

The following figures present the system structure in Warnier diagrams.

- Inventory Control- Figure 6 shows an overview of the highest level of the control structure. The main functions of the system are presented in the second level of the tree:

1. Maintain Inventory File
2. Maintain Vendor File

3. Enter Transactions
 4. Process End of Period
 5. Backup Inventory Files
 6. Produce Reports
- Perform Inventory Control- Figure 7 provides a skeleton of the sequence of program functions.
 - Maintain Inventory Item File- Figure 8 shows the control structure for maintenance of the item file.
 - Maintain Vendor File- Figure 9 shows the control structure for maintenance of the vendor file.
 - Enter Transaction- Figure 10 shows the control sequence of obtaining transaction information from the user and entering the transaction record into the transaction file.
 - Process End of Period- Figure 11 shows the sequence of events undergone during the end of month activities.
 - Backup of Inventory Files- Figure 12 shows the file backup process.
 - Produce Reports- Figure 13 documents the dialog with the user which selects and produces reports.

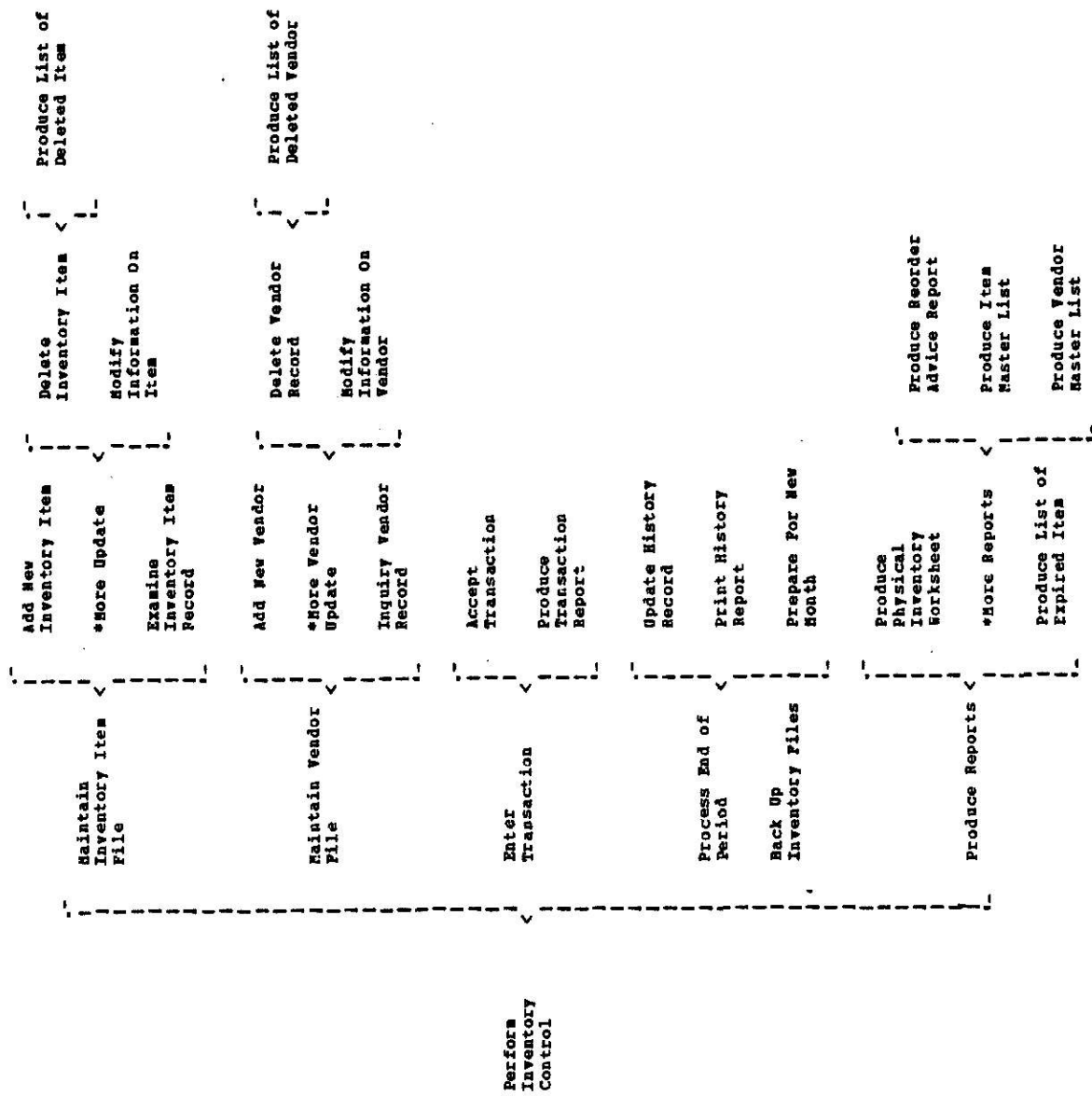


Figure 6: Inventory Control Overview

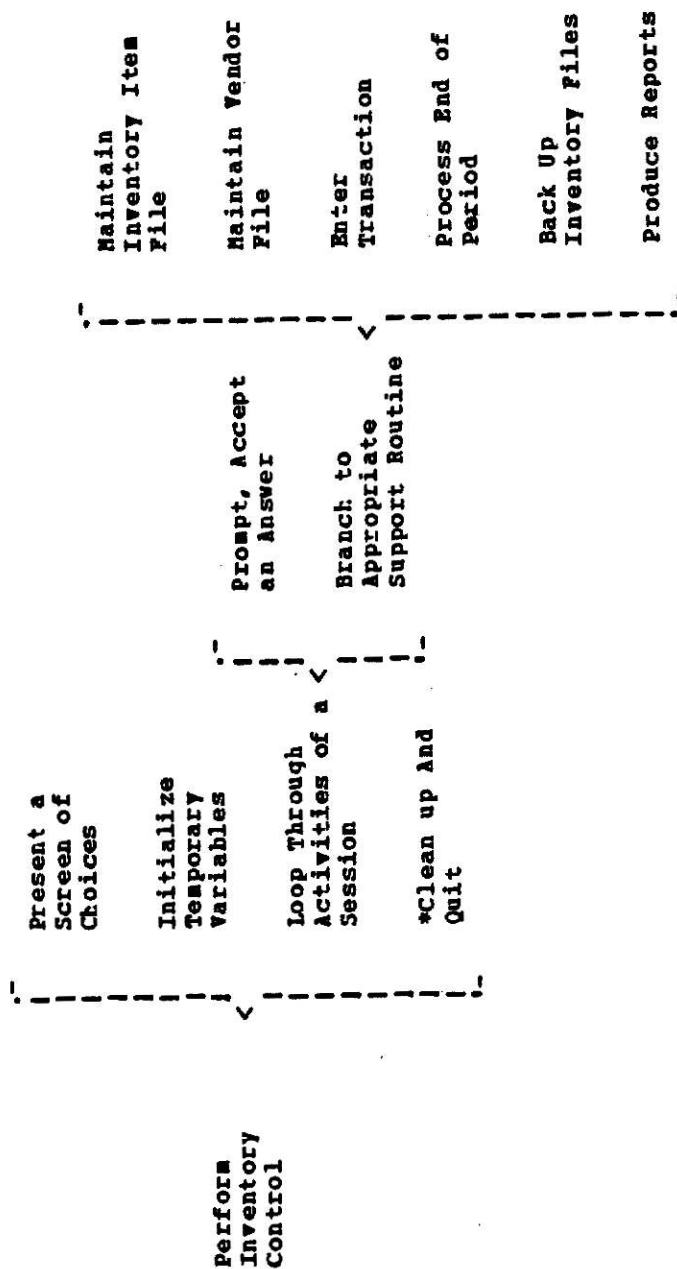


Figure 7: Perform Inventory Control

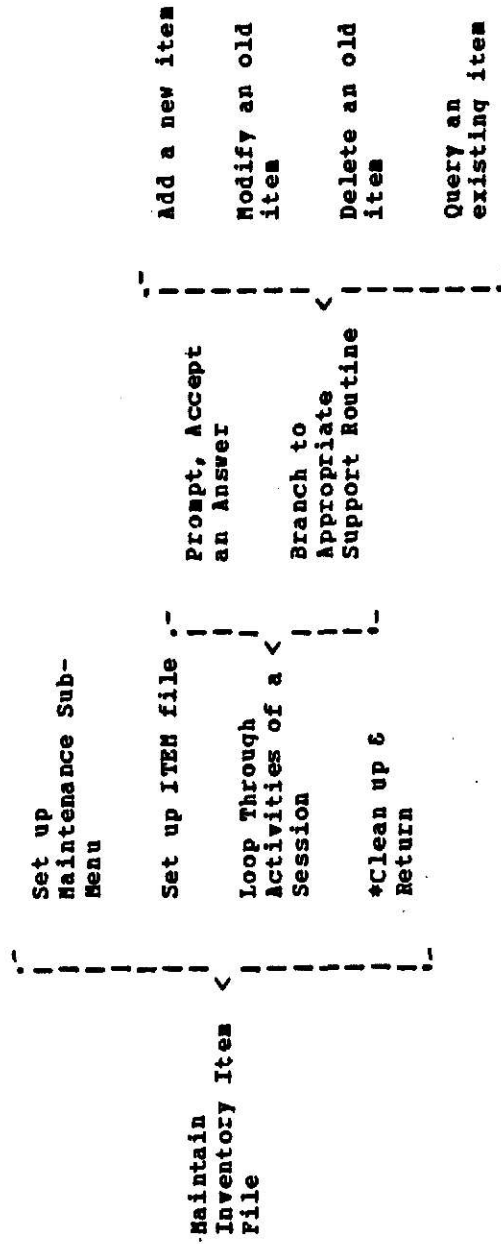


Figure 8: Maintain Inventory Item File

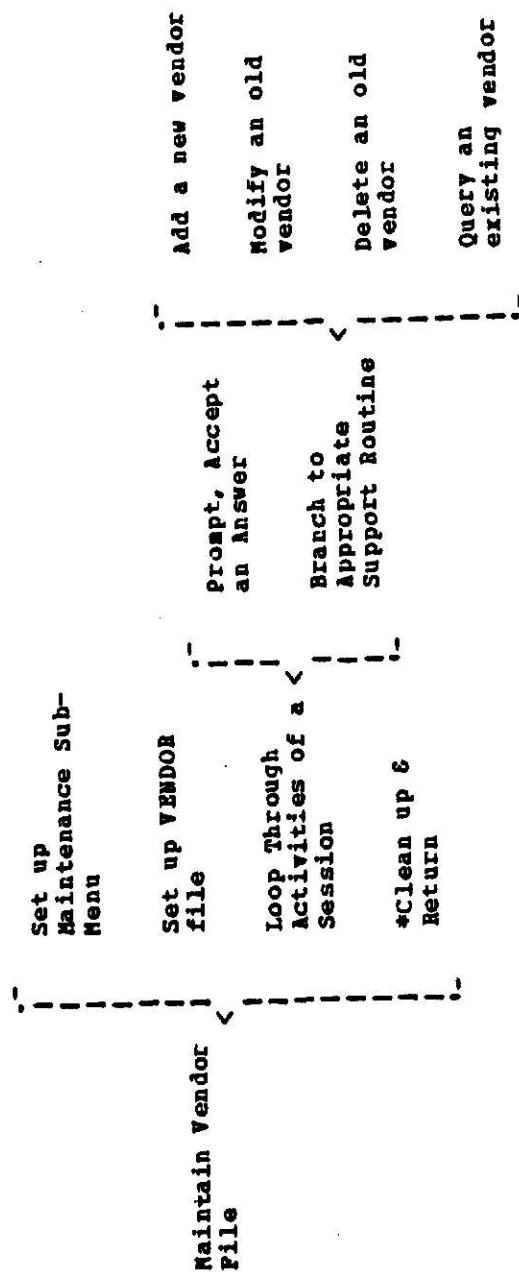


Figure 9: Maintain Vendor File

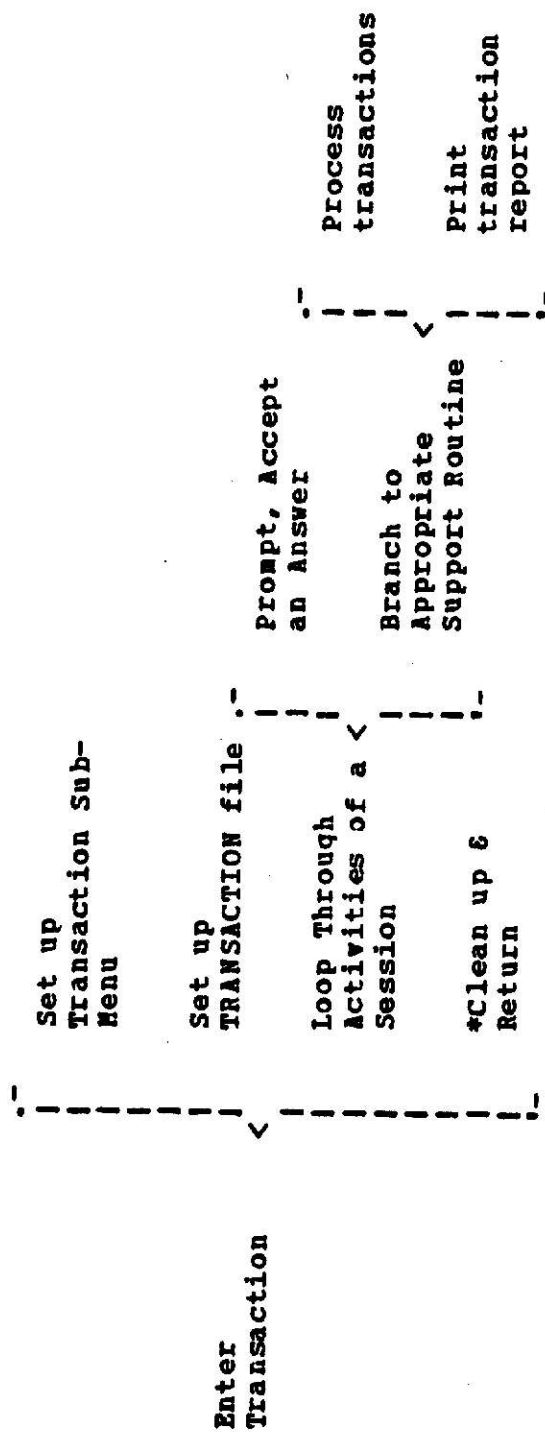


Figure 10: Transaction Module

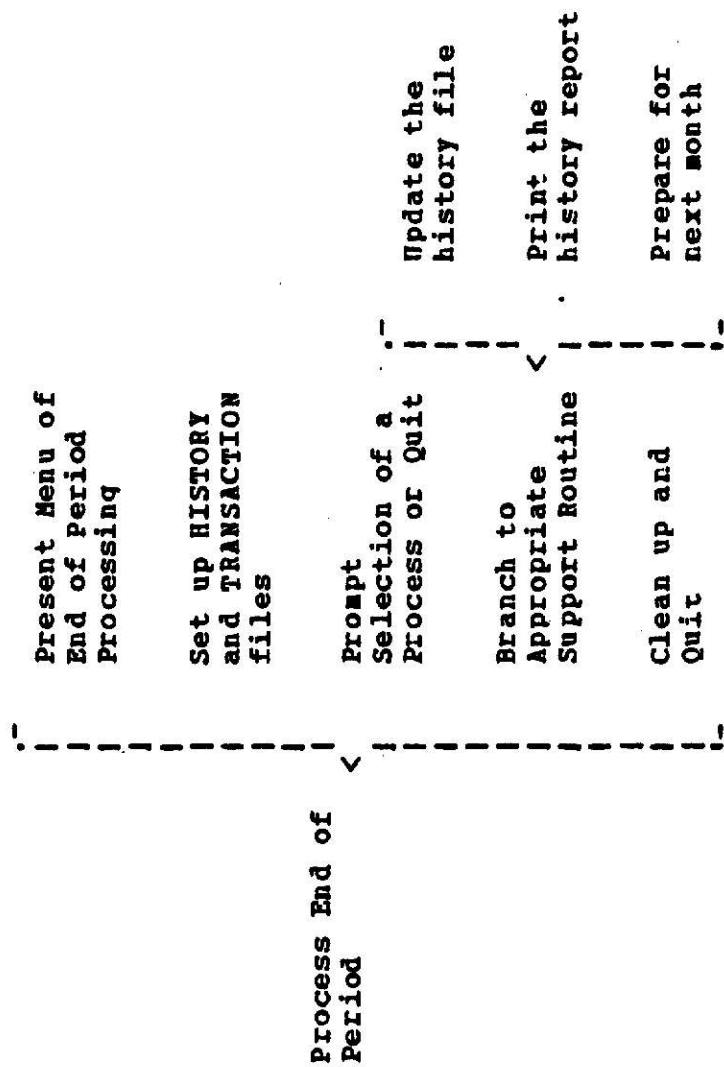


Figure 11: Month End Processing Module

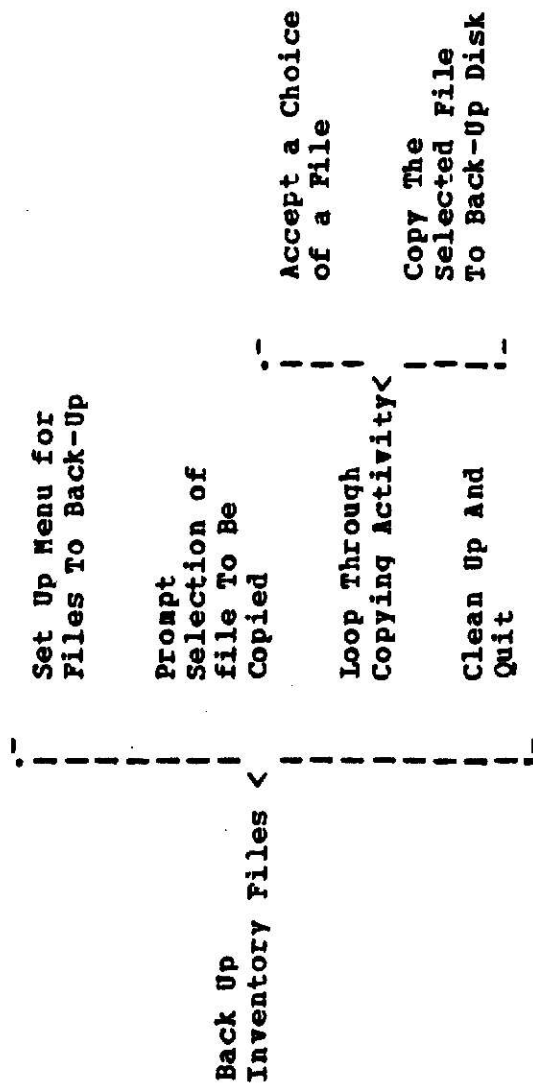


Figure 12: Back-up of Inventory Files

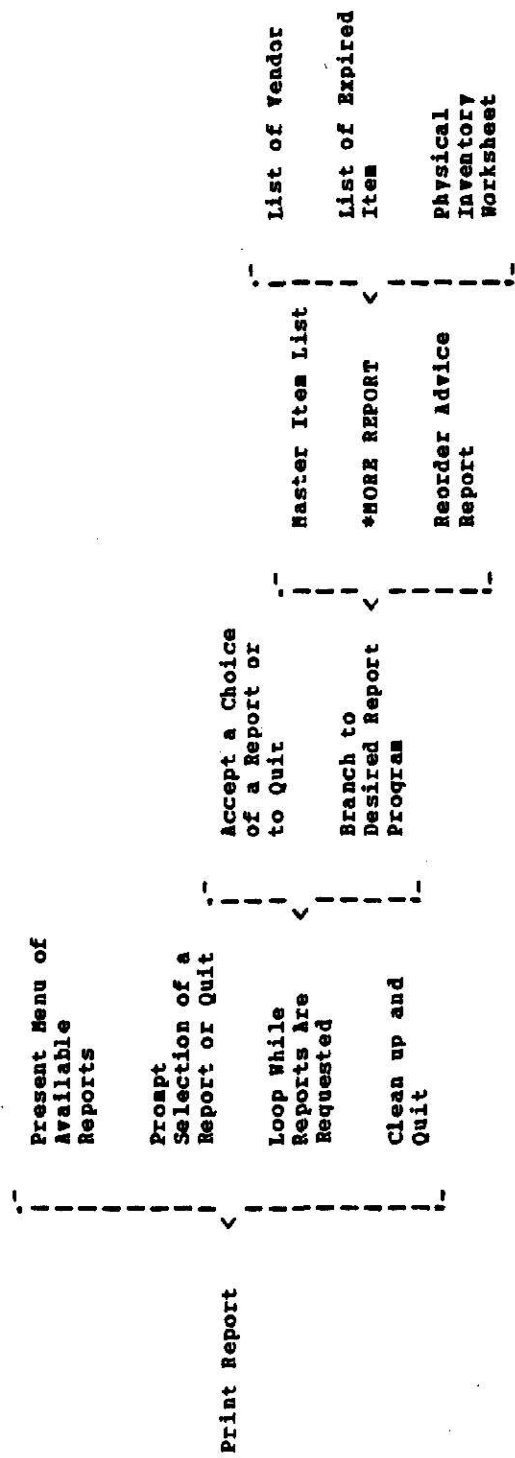


Figure 13: Inventory Report Selection

3.2.2 Data Structure

Three principal database files: the ITEM file, the HISTORY file and the TRANSACTION file in the inventory management system. The files were defined as part of the output oriented design method. The output reports were analyzed and each data element represented by the report was identified. The data elements were grouped to form the system files. Generally one report is associated with each file. This is commonly the case in systems development (8). The elements in the database are then analyzed and determined to be either input or derived variables. The ITEM file is the file which maintains the quantity on hand, both period-to-date and year-to-date usages, and other necessary information of item. The HISTORY file contains item code number, both period-to-date and year-to-date usages of each department within the hospital and the total amount spent. The TRANSACTION file is the file which keep a record of all transactions entered into the system in that period. The TRANSACTION file is the basis for the production of the monthly history report.

All the files are indexed on a key field. For example, the ITEM file would be indexed on item code, the CUSTOMER file is indexed on customer code and the VENDOR file is indexed on vendor code. Indexing provides a way of keeping the records of a file in a specified logical order without sorting them. This allows access to individual records with

a certain attribute without having to do a complete scan of the file and allows access to the records in a specific order without sorting them.

In addition to these major files, two support files exist. The CUSTOMER file and the VENDOR file are used only in connection with the monthly or on-demand reports. The CUSTOMER file provides part of the title for the reports and the VENDOR file provides the information about vendor in connection with reorder advice report.

When out-of-date transaction records are deleted to decrease file size and increase speed of processing, the deleted records appear in the OLD TRANSACTION file.

The Warnier diagram representation of the record structure of these five files is presented below.

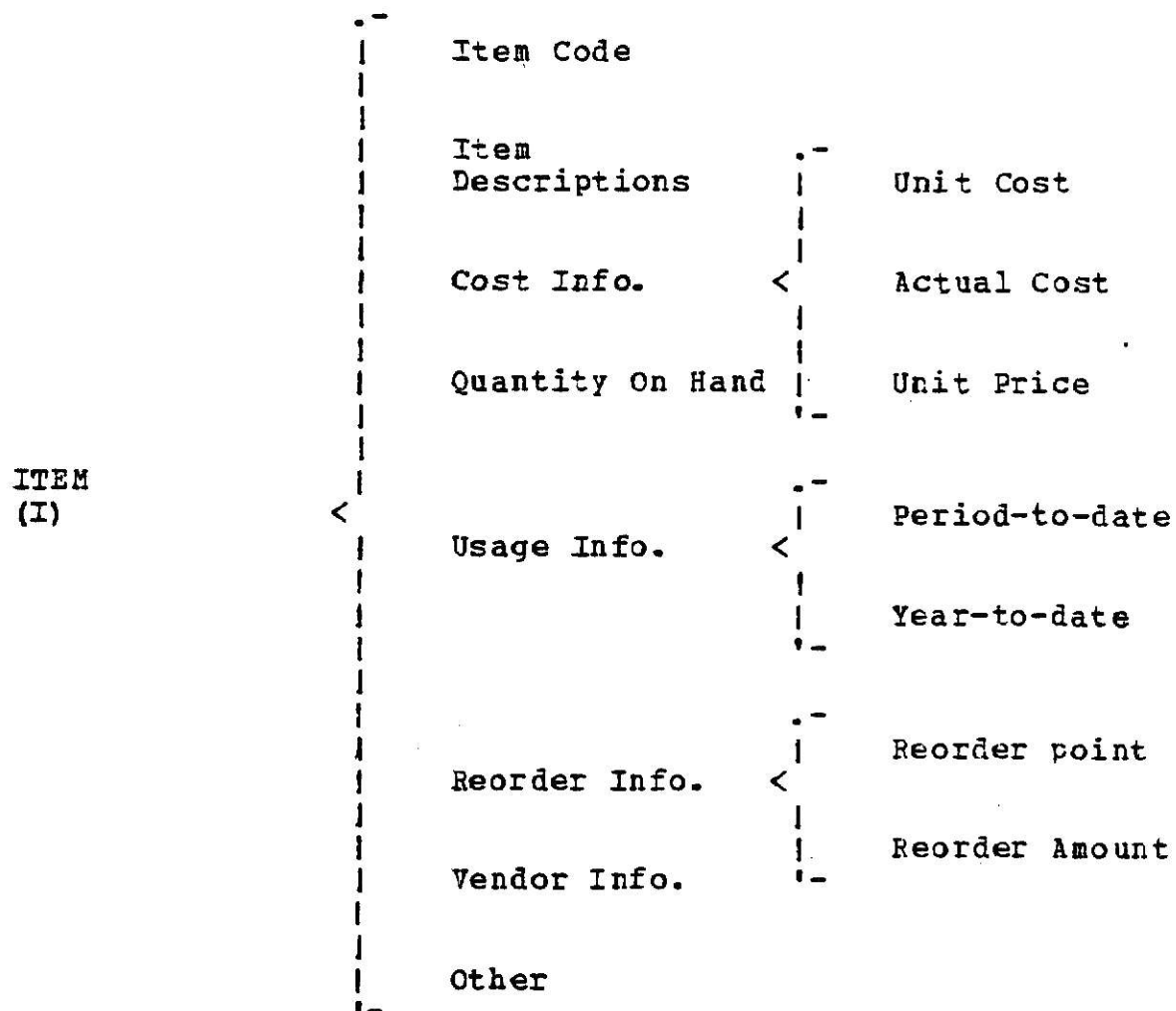


Figure 14: Structure of Item File

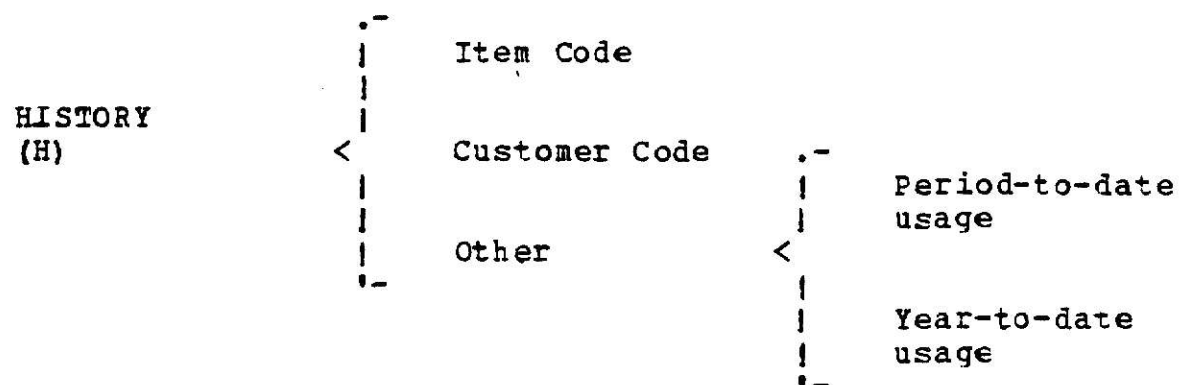
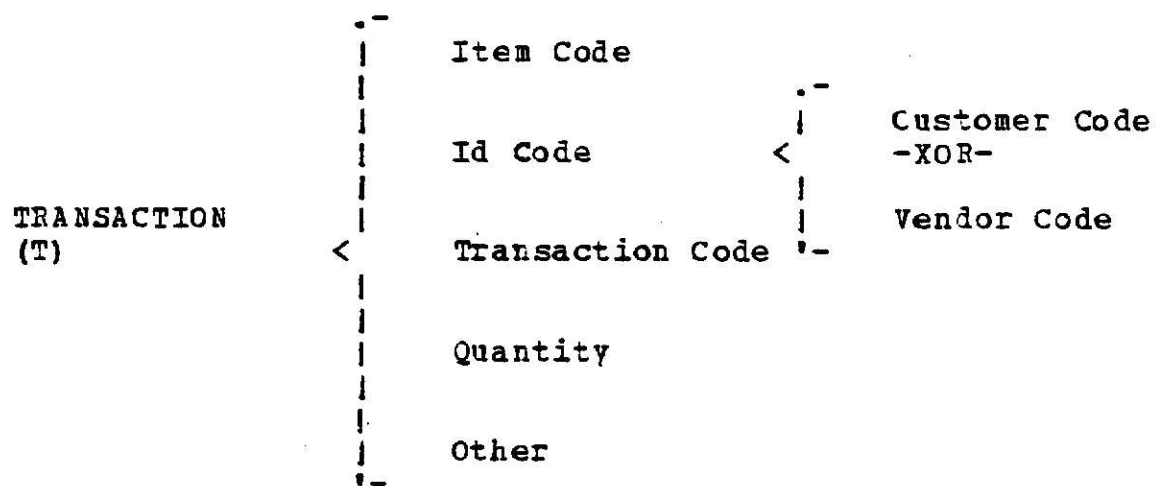


Figure 15: Structure of History File



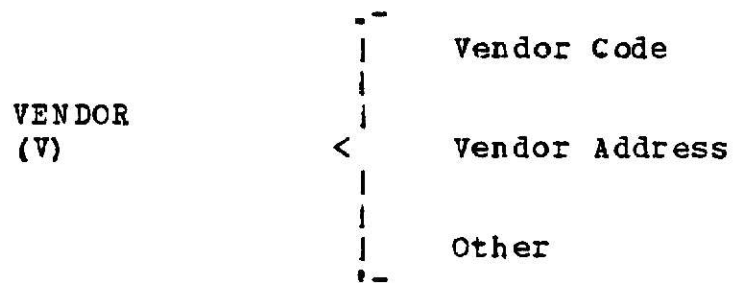


Figure 17: Structure of Vendor File

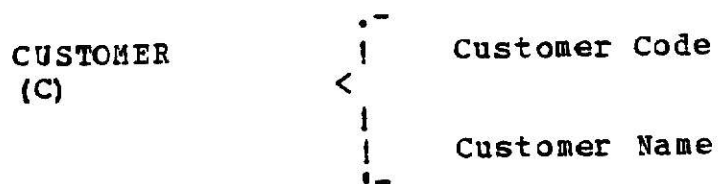


Figure 18: Structure of Customer File

These five files taken together form the inventory control system database. The logical arrangement or schema of the database (DBTG Definition) is shown in the figure 19

(12)

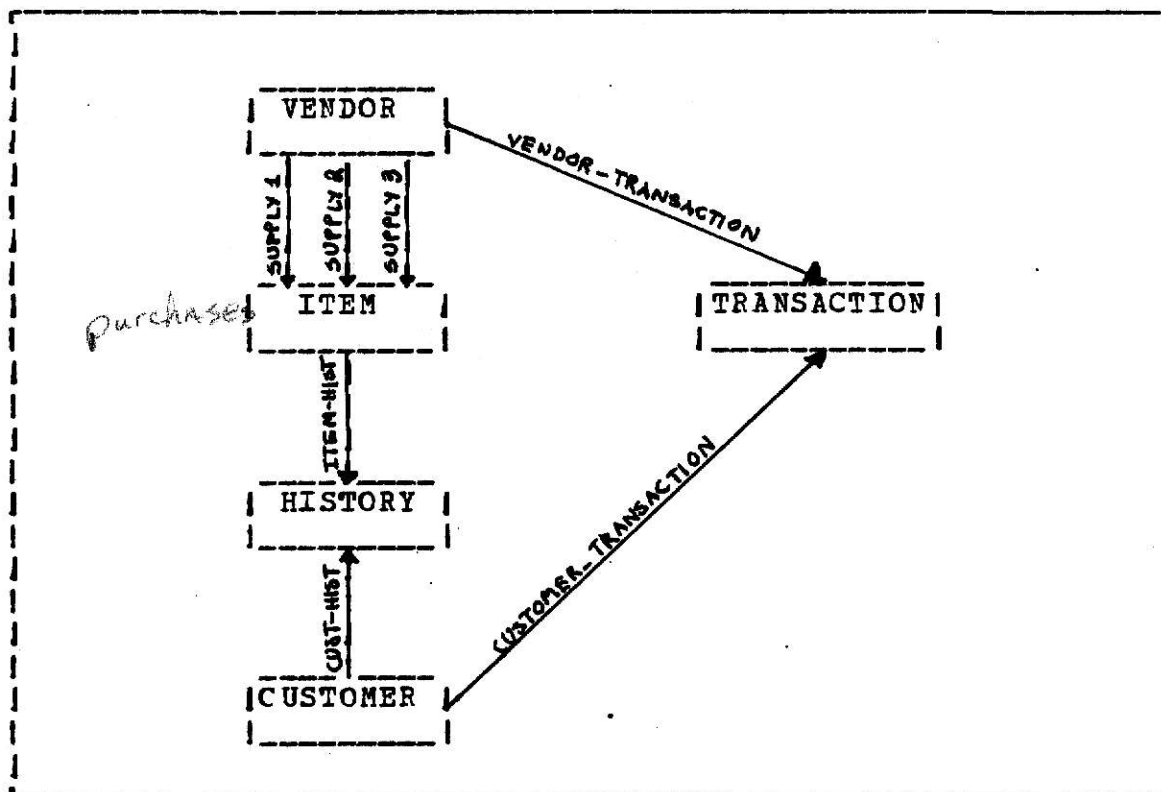


Figure 19: Database Schema

• Record Descriptions:

1. Vendor - contains information on a vendor which supplies one or more item to the stockroom.
2. Item - describes items which may be supplied in the stockroom.

3. History - serves as a junction records to implement M:N relationship between customers and items. Any number of customers may receive any number of items.
 4. Customer - describes customers receiving items from the stockroom.
 5. Transaction - records movement of items in and out of the stockroom. Item come in from vendors and go out to customers.
- Set Descriptions:
 1. Primary Supply - links a vendor record to the item from which the last order of items was obtained.
 2. Alternative Supply1 and Alternative Supply2 - provides information on two other vendors from which an item may be obtained.
 3. Item History and Customer History - implements the M:N relationship between customer and item.
 4. Vendor Transaction - links a vendor to a transaction.
 5. Customer Transaction - links a customer to a transaction.

Chapter IV

IMPLEMENTATION

The programming language/environment which was selected to host the inventory management system on microcomputer is dBASE II, a relational database management system from Ashton-Tate which provides excellent programming capability as well as good native facilities for management of data on microcomputer. The system is available on a wide range of microcomputers and has a programming language designed to make the system into an application-development-system. Also, dBASE II is well-documented and supported, and uses a block-structured language in which modular and highly readable code is produced.

One major advantage of the use of dBASE II as the host system for the inventory control is that dBASE II is designed as a user-friendly data management system. Its commands are well named to be indicative of their function and the manual which describes the dBASE II system is quite clear. Also, dBASE II can be regarded as a query/report language used to access the inventory information being maintained by the inventory management system.

The restriction arising from the use of dBASE II are the limitations of 1) number of fields per record is 32; 2) max-

imum record size is 1024 bytes; 3) number of files open simultaneously is two; 4) and number of records per file is 65535.

The system, as currently implemented, does not require any interfaces outside of the dBASE II environment. The user communicates with the databases through interactive menus or screen. Basically each screen is associated with a particular function and therefore is associated with a particular module.

4.1 SYSTEM FILES AND DESCRIPTIONS

In the inventory control system, the following data files are used and all of them are stored on the harddisk drive of a microcomputer.

4.1.1 Item Master File

This file maintains the item's quantity-on-hand, both period-to-date and year-to-date usages, and other necessary information. Each record contains the item code, description, unit cost and price, reorder detail, vendor detail and usages detail. These records are indexed on item codes. The structure of record is presented in figure 20

FIELD	NAME	TYPE	WIDTH	DEC	DESCRIPTION
01	Item:code	C	5		Identification
02	Location	C	4		Location of item
03	Item:Name	C	30		Name of item
04	Item:Desc	C	30		Description
05	Generic	C	30		Generic name
06	Drug:Code	C	1		Scheduled code
07	Disc:Code	C	1		Discount code
08	Act:Cost	N	6	2	Actual cost
09	Nor:Cost	N	6	2	Normalized cost
10	Ret:Cost	N	6	2	Retail cost
11	Quantity	N	5		Quantity on hand
12	Ptd-Usage	N	5		Period-to date
13	Ytd-Usage	N	7		Year-to-date
14	Reord:Pt	N	3		Reorder point
15	Reord:Amt	N	3		Reorder amount
16	Date:Ord	C	6		Date ordered
17	Dlastrec	C	6		Date last received
18	Dexpire	C	4		Expiration date
19	Vitem:No	C	5		Vendor item code
20	Nat:Code	C	10		National drug code
21	Buy:Code	C	1		Purchase code
22	St:Code	C	1		State commodity code
23	Vendor1	C	5		Vendor Id
24	Vendor2	C	5		Vendor Id
25	Vendor3	C	5		Vendor Id

Figure 20: Structure of Item Record

4.1.2 Transaction File

Purchase
or sales

→ historical
record
49

This is a file containing inventory postings, one record per posting. The sequence of the file is the order in which the postings were made. Each posting contains the item code, transaction code, vendor id or customer id, posting date, the quantity and total amount of the item to which the posting were made. See the structure of transaction record in figure 21

FIELD	NAME	TYPE	WIDTH	DEC	DESCRIPTION
01	Tran:Code	C	2		Transaction code
02	Item:Code	C	5		Item id
03	Tran:No	C	6		Transaction no.
04	Id:No	C	5		Customer/Vendor id
05	Qunatity	N	4		Number of item
06	Amt	N	7	2	Dollar amount
08	Upd:Flag	C	1		Update flag
09	Remarks	C	20		Comments

Figure 21: Structure of Transaction Record

4.1.3 History File

This file contains item code number and customer codes. Each customer code contains period-to-date usage, year-to-date usage and total amount spent this year. See the structure of history record in figure 22

From the data structure design, the history record would contain a key to one item record and one customer record.

FIELD	NAME	TYPE	WIDTH	DESCRIPTION
01	Item:Code	C	5	Item identification
02	Anphr	C	30	Customer 1
03	Anpht	C	30	Customer 2
04	Ansci	C	30	Customer 3
05	Anshl	C	30	Customer 4
06	Cenpr	C	30	Customer 5
07	Clpth	C	30	Customer 6
08	Dilab	C	30	Customer 7
09	Kabsu	C	30	Customer 8
10	Ladsk	C	30	Customer 9
11	Laegp	C	30	Customer 10
12	Lafap	C	30	Customer 11
13	Lafsv	C	30	Customer 12
14	Lasur	C	30	Customer 13
15	Lbmed	C	30	Customer 14
16	Patho	C	30	Customer 15
17	Radio	C	30	Customer 16
18	Saden	C	30	Customer 17
19	Saemr	C	30	Customer 18
20	Saicu	C	30	Customer 19
21	Samed	C	30	Customer 20
22	Sapth	C	30	Customer 21
23	Sasur	C	30	Customer 22
24	Sawds	C	30	Customer 23
25	Smarf	C	30	Customer 24
26	Sszoo	C	30	Customer 25
27	Toxic	C	30	Customer 26
28	Misc1	C	30	Special project1
29	Misc2	C	30	Special project2
30	Misc3	C	30	Special project3
31	Misc4	C	30	Special project4
32	Misc5	C	30	Special project5

Figure 22: Structure of History Record

Since the population of customers in this implementation is limited, information on all customers is contained in each item record. This saves lookup time. Since there is only one record per item, the history file only has to be indexed on one key (item:code). If implemented as a 3NF relation as would normally be done the record would take the following form.

Item-code	C	5
Cust-code	C	5
Ptd-usage	N	6
Ytd-usage	N	9
Total-amt	N	10
TOTAL:		35 bytes

For 2000 items and 31 customers the file would take 2.17 megabytes and indexes on item code plus customer code would have to be maintained simultaneously.

As currently implemented the file with 2000 items would take 1.56 megabytes and has to be indexed only by item code. The dBASE II restriction of 32 fields per record made it necessary to store multiple logical pieces of information in one physical field. The period-to-date and year-to-date usages, and total amount spent fields are kept concatenated together in character form in a single field. These are separated by use of the substring operation, converted to numeric, where needed, and manipulated as necessary. The overhead of this conversion of data type is not noticeable

when compared with the record search time involved with manipulating large files indexed on multiple keys.

4.1.4 Customer File

This file contains customer code and customer name. The file is used for report printing purpose only. See the structure in figure 23

FIELD	NAME	TYPE	WIDTH
01	Cust:Id	C	5
02	Cust:Name	C	40

Figure 23: Structure of Customer Record

4.1.5 Vendor File

This file contains the information about vendor in connection with the reorder advice report. See figure 24 for the structure of vendor record.

FIELD	NAME	TYPE	WIDTH
01	Vendor:Id	C	5
02	Name	C	30
03	Address	C	30
04	City	C	20
05	State	C	02
06	Zipcode	C	09
07	Phone	C	12
08	Contact	C	30
09	Pur:Id	C	10

Figure 24: Structure of Vendor Record

4.2 SYSTEM MODULE AND SCREEN DESCRIPTIONS

In this section, details of the operation of each module which produces screens and appearance of each screen is discussed briefly. A description of each of these screens is presented below. Their calling structure is presented in figure 25 Sample system reports are included in the appendix section.

- Perform Inventory Control is the main control program of the inventory system. The program is initiated by starting the microcomputer. The password routine is executed, if the operator then enters the correct password access to the inventory system will be gained. The main menu is displayed, the operator now has a choice of operation to select. The alternatives are listed as options. (See figure 26)

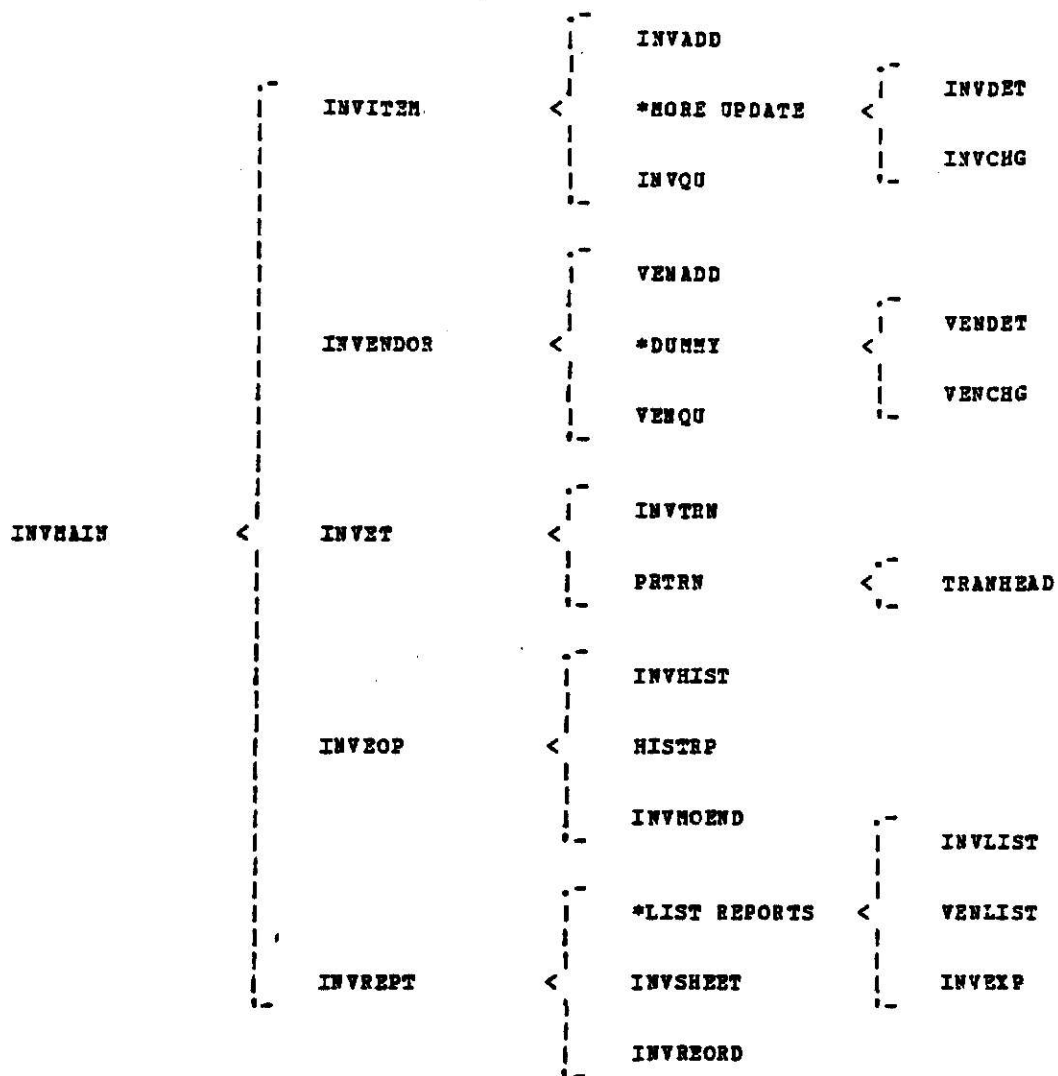


Figure 25: System Calling Tree

```

                                INVENTORY CONTROL VERSION 1.0
-----
YOU MAY SELECT ANY OF THE FOLLOWING PROGRAMS

MF... Master File Maintenance
VF... Vendor File Maintenance
ET... Enter Inventory Transactions
EP... End of Period Update & Report
RP... Inventory Reports (Submenu)
BK... Back-up Files Routines
EN... End System Select (Return to OS)
-----
WHICH PROGRAM DO YOU WISH TO SELECT?

```

Figure 26: Main Inventory Menu

- Main Menu Selection "MF Master File Maintenance": The master item file maintenance programs provide facilities for adding new items, or deleting, changing or querying existing items. The programs are cyclic, that is, you can do maintenance to as many items as necessary once the programs have been started.

```

                                MAINTENANCE ITEMS FILE

AD... Add a New Item
DE... Delete an Existing Item
CH... Change an Existing Item
QU... Query an Existing Item
RE... Return To Main Menu

```

Figure 27: Master File Maintenance

When additions are made to the item master file, the new record is added and placed in proper position before other processing takes place for the new items. A check to assure non-duplication of item number is made before it is added. If the item is already existed , the message "Duplicate Item Number: Try Again" will be displayed. To add new items, enter the selection 'AD' (see figure 27) then enter the item number to be added. The program "INVADD" will then prompt the operator to enter the item descriptions.

Change of an existing item can be accomplished by entering the item number to be changed. The program "INVCHG" will display the item's information. Now enter a reference number in front of the field you wish to change and the field's updated value. The record will be rewritten to the item master file. Note that the period-to date and year-to-date usages fields can not be changed since these fields are generated by the program.

To delete an existing item, enter the selection 'DE' (see figure 27); the program "INVDET" (see figure 25) will prompt for item to be deleted. Then the program will display the item name and descriptions of that item for which a deletion was requested.

When items are deleted, the item record is flagged for delete then the record is copied to the deleted

item file. The system will prompt the operator to start the printer and print a list of deleted items for the audit trail. When the deleted records are to be physically removed from the item master file, the system will display the following messages: "There will be a delay while the item file is packed". The time required for reorganizing the item master file is dependent on the size of the file. A list of deleted item report is also provided as a permanent record for checking.

Either adding, deleting, or changing an item, the system will prompt for ACCEPT(Y/N). The response 'Y' is entered if the item shown is correct, or 'N' is entered if the item shown is incorrect and aborting the transaction is desired.

Query an Existing Item "INVQU" (see figure 25). This program provides information on the item through the input of item number. The program accesses the ITEM file and display the item information otherwise the "No Such Item Number" message is displayed. The program then loops back to the item number entry for next item number. This feature is provided to save look-up time in listings for ease of determining the status of the item.

Note that the item maintenance programs do not allow the operator to change item quantity "on hand" values.

This "on hand" may only be changed through transaction postings (selection 'ET' in main menu see figure 26) with a record of the transaction printed on a posting report. This provides an audit trail for postings and insures that the item on hand will be equal to the sum of the value of transactions posted and any previous quantities.

If item number '99999' is given as input in "INVADD", "INVCHG", "INVDET" or "INVQU" the program will return the control to "INVITEM" program. (See figure 25)

If the selection 'RE' is selected (see figure 27), the program "INVITEM" will display the main menu and the control is returned to "INVMAIN".

- Main Menu Selection "VF Vendor File Maintenance": The vendor file maintenance module "INVENDOR" provides basically the same facilities for adding new vendors "VENADD", or deleting "VENDET", modifying "VENCHG" or examining "VENQU" existing vendors as the ITEM file maintenance. See above section for details.
- Main Menu Selection "ET Enter Transaction": If transaction input module is requested, the system will provide a transaction menu (see figure 29). To initiate the program; enter 'SA' for order processing; 'RE' for merchandise receipt; 'PO' for purchase order processing; 'AD' for adjustment transaction or 'RE' to return

<p>MAINTENANCE VENDORS FILE</p> <p>AD... Add a New Vendor DE... Delete an Existing Vendor CH... Change an Existing Vendor QU... Query an Existing Vendor RE... Return To Main Menu</p>
--

Figure 28: Vendor File Maintenance

to main menu. (See figure 29) After the selection is made, the item is called through the input of a item number. The program is called from the Perform Inventory Control "INVMAIN". (See figure 25) The program "INVTRAN" accesses the ITEM file and displays the matching item description. Verification is requested on a prompt.

<p>INVENTORY TRANSACTION MENU</p> <p>SA... Enter a Sale Item AD... Enter an Adjustment PO... Post an Order To Vendor RI... Enter a Receipt Item PR... Print Trans Report RE... Return To Main Menu</p>
--

Figure 29: Transaction Menu

If order processing is selected, the customer number, the transaction number, and the quantity order are entered by the operator. The program display the unit price and calculate the total amount. The program then adds the transaction to the TRANSACTION file and loops to item number entry for another transaction.

If merchandise receipt is selected, the purchase order number, the vendor number and the quantity receipt are entered. The program updates the quantity on hand, the date of last received on the ITEM file and loops back for next transaction.

If purchase order is requested, the purchase order number, the vendor number and the quantity order are entered. The program updates the date of order. This process is repeat until all purchase orders have been entered to the system.

If inventory adjustment is selected, the transaction number, the quantity adjust are entered. The program then calculates the total amount of adjustment and display. The comment field is available for reason of adjustment whether it is a return, breakage, damage or theft. The program updates the quantity on hand and loops back for next item.

Normally with correct data input, the transaction is added to the file and the response "Y" will automatically loop back to the item number entry. Otherwise

the program will disregard the transaction and the operator can reenter the data.

All transactions above are repeated until item number '99999' is used. The program then returns control to "INVT" program and another selection can be made.

At the end of business day, the transactions should be printed for auditing purposes. Print Transaction "PRTN" produces the daily transaction report. The operator is given an opportunity to start the printer and align the form. The notes that the program will take a few minutes to complete is displayed. The transaction report is broken down into four parts: sale transaction; receipt of item from vendor; item on order and adjustment transaction, and the summary report is produced to show the total of sale and adjustment for the day.

After the transaction report is printed, all transactions in the TRANSACTION file are added to the monthly TRANSACTION file for month end processing and deleted to clear for the next day's transaction.

- Main Menu Selection "EP End Of Period Processing":

This module "INVEOP" focuses on the proper maintenance of the accounting system at the end of the accounting period. At the end of accounting month, some bookkeeping must be done. Before month end processing can be started, all postings for that day have been completed,

all dialy activities have been generated and approved, and that no further activity for that day is required. A monthly TRANSACTION file copy of the data diskette(s) should be made and retained for back-up purposed.

```
MONTH END PROCESSING MENU

UH... Updating History File
HR... History Report
MO... Month End Processing
RE... Return To Main Menu
```

Figure 30: End of Period Menu

This program "INVEOP" provides a choice of updating the HISTORY file, printing history report, preparing files for next month or return to main menu (see figure 30). The "INVMAIN" is passed the control to "INVEOP" when the end of period processing is selected from main menu. The update history file program is initiated by selecting "UH" from figure 30

Updating History File "INVHIST": The program updates the HISTORY file. The monthly TRANSACTION file is an input file used to update the current period sales history, the program reads the customer identification from the sale transaction record and locates that customer on the HISTORY file then updates the

period-to-date and year-to-date usage fields. The period-to-date total amount is calculated and added to year-to-date total amount field on the HISTORY file to ensure that the current unit price is used for that period. The program loops on this TRANSACTION file until all sales transaction have been processed and the month end flag is set for each processed record. The program runs without operator participation except for printer preparation. The control of program is returned to INVEOP when the process is completed.

Printing History Report "HISTRP": This program provides the history report of selected customer or all customers. Normally the history report contains information for all customers for document. The report is printed by each customer and by item numbers. The total sales of each customer is provided for financial control.

Preparing for New Month "INVMOEND": This program provides the business function of closing the current period so that once a final history report has been produced, the system will not permit additional entries to that accounting period to distort the financial record. The period-to-date usage field on the HISTORY and ITEM files are zeroed out for next period. The monthly TRANSACTION file is offloaded to backup copy of TRANSACTION file to decrease the monthly TRANSACTION

file size and increase the speed of processing at month end.

- Main Menu Selection "RP Report Sub-Menu": When the report menu "INVREPT" is selected, a reports sub-menu is presented. (See figure 31)

<p style="text-align: center;">INVENTORY REPORT MENU</p> <p>IC... Item Catalog Report LV... List of Vendors PI... Physical Inv. Worksheet EX... List of Expired Drugs RA... Reorder Advice Report RE... Return To Main Menu</p>

Figure 31: Report Menu

In all reports, the user is given an opportunity to start the printer and align the paper before the printing of the report is started. Report printing has no side effect on the condition of the files; reports in this menu can be obtained repeatedly without altering the file contents in any way. See sample reports in Appendix.

- Main Menu Selection "BK Back-Up File Maintenance": If back-up inventory files is selected, the program "INVBACK" will provide facilities to copy a specified file. The file can be specified by entering a selection from sub-menu screen. (See figure 32)

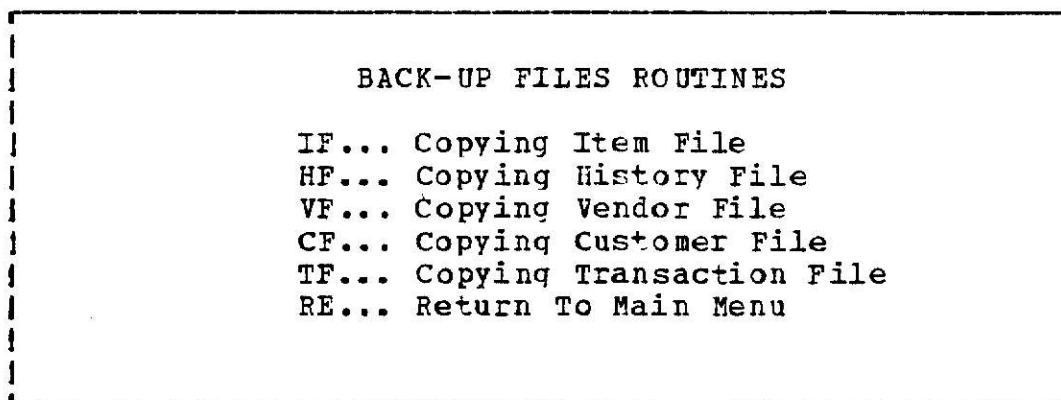


Figure 32: File Backup Menu

After the selection is made, the system will prompt the operator to insert back-up disk in the diskette drive. The data can be shared one or more file(s) to the same diskette, it depends on how large the data files and disk storage capacity is. After the copying process of that file is finished, the system will loop back and prompt for another file to be back-up or quit and return to main menu screen.

- Main Menu Selection "EN End System Selection": The final section provide mean to exit from the inventory management system to the microcomputer operating system.

An additional set of programs, Heading1, Heading2, Heading3, Transhead, Histhead, are supported programs which contain the title heading of report. The last program is the security password routine. Before any processing can be performed, the user inputs a security

code with control access to the inventory system. The program restores memory variables from a constant memory file and matches the user entered code. The program allows three tries to match the memory variable code. If the operator can not match the code, the program will display a warning message of unauthorized personnel and return to the operating system. The program then saves the number of tries of previous access to memory variable. The authorized user then have to reset the password routine to be able to access to the inventory system. However, anyone with microcomputer and dBASE II knowledge can still obtain access to system data files.

Chapter V

SUMMARY AND FUTURE WORK

The system defined and implemented as described above resulted in an operational automated aid to inventory tracking and maintenance. The system is implemented in a language which is portable to a wide variety of micro computers and a number of popular operating systems. The system is single user system requiring only a 16 bit word microcomputer with 128K of memory with terminal and a printer to be operational.

The resulting system allows the user to keep track of vendors and customers, and the items supplied and consumed. The user is able to add, delete and modify information on vendors and items. The user enters information on transactions which track the movement and supply of inventory items. The system is able to alert the user when reorder levels have been reached and provide information on the suppliers of items. In addition the system provides information on the consumption of items by customer and a list of items which must be discarded because they have expired.

The input and reporting requirements have been met. All input is entered interactively and all nonterminal output is sent automatically to the printer. In some installations

the ability to read files containing batches of transactions may be appropriate. Since this is a single-processor implementation, use of this system to monitor the inventory of a distributed stockroom may require recording the transactions on diskette followed by transport of the diskettes to the site of the database. Inventory reports subsequently could be returned to the various stock locations in machine readable form on diskette.

Further analysis of the inventory data such as stock flow and prediction could be done on summary files, another possible extension to the existing system. Any number of analyses may be performed on machine-readable output from the system.

Added system flexibility would be found through implementing the system in a multiuser environment. This may be done either by reprogramming the current system to be compatible with multiple machines communicating on a local area network, requiring a file or record lock facility to be utilized, or the system may be reimplemented in another database language which has built-in facilities for handling multiple users concurrently, as well as on multi-user hardware systems.

REFERENCES

1. Bergland, Glenn D. and Gordon, Ronald G. Tutorial: Software Design Strategies New York: IEEE, 1979.
2. Charles, Biggs L. Managing the Systems Development Process Englewood Cliffs: Prentice Hall Inc., 1980.
3. Conrow, Kenneth. "The Automated Warnier Diagram Production" Manhattan KS: KSU Computing Center, 1984
4. Donaldson, Hanish. A Guide to the Successful Management of Computer Project New York: John Wiley & Son, 1978.
5. Haveisen, William D. and Camp, James L. Business Systems for Micro-Computers Englewood Cliffs: Prentice Hall Inc., 1982.
6. Jackson, M. A. Principals of Program Design New York: Academic Press, 1975.
7. Jackson, M. A. "Constructive Methods of Program Design," New York: Spinger-Verlary, 1976.
8. Langston, Kitch. Structures Information Topeka: Langston Kitch & Associates, 1978.
9. Orr, Kenneth T. "Introducing Structure Systems Desgin," New York: IEEE, 1979.
10. Orr, Kenneth T. Structured Requirement Definition Topeka, Kansas, 1981.
11. Pressman, Roger S. Software Engineering: A Practitioner's Approach McGraw-Hill Inc., 1982.
12. Ross and Schouran. "Structured Analysis for Requirements Definition," Classics in Software Engineering New York: Yourdon Press, 1979.
13. Tsichritzis, D. C. and Lochovsky, F. H. Data Base Management Systems New York: Academic Press, 1977.
14. Warnier, Jean and Dominique. The Logical Construction of Programs New York: Van Nostrand Reinhold, 1976.
15. Yourdon, E. N. and Constantine Larry L. Structure Design Englewood Cliffs: Prentice Hall, 1979.

Appendix A

USER'S GUIDE

A.1 INTRODUCTION TO INVENTORY SYTEM

The Inventory System provides facilities to maintain records of items in the stockroom. New items may be added and discontinued items may be deleted as necessary. Postings may be entered daily and a dially transaction report is printed for all postings. The system provides for a current quantity-on-hand, period-to-date usage and year-to-date usage of each item to retain information required for detailed end of the month financial reporting. Transactions are retained for one year to provide a year-to-date history report.

A.1.1 System Highlights

- User definable item code structure of up to 5 digits.
- System capacity is 65,535 records per file. The accumulated year to date transaction file is expected to be the largest file in the system. This is sufficient to allow most businesses to carry a full year of transactions on-line.
- Interactive, operator oriented posting of transactions and file maintenance through the video display screen. The operator is given guiding instructions on the screen for each enry.
- Detection and correction of common errors as the data is being entered. Extensive checking on item code is done on each entry and correction is required immediately.
- All reports normally required for inventory control processing are automatically prepared on a microcomputer printer. Samples of all system reports are included in the Sample Reports Section. System reports include:
 1. Daily Transaction Report
 2. List of Deleted Items & Vendors Report

3. History Report
 4. Reorder Advice Report
 5. Item Catalog Report
 6. Vendor Catalog Report
 7. Physical Inventory Worksheet
 8. List of Expired Items Report
- The system is written in dBASE II making most program alterations a simple task.
 - The indexing technique permits information to be quickly retrieved and displayed on the screen.
 - A fully menu driven system. All selections are initiated from the Main Inventory Menu Screen and return to that screen when complete.

A.2 THE ACCOUNTING CYCLE

In this section, the traditional accounting cycle is reviewed. Note that the use of the automated system does not eliminate any of the traditional steps. The advantages of an automated system are:

- Substantial time saving throughout the cycle resulting in earlier presentation of financial reports.
- Accuracy and consistency is maintained at all times.

The following check list may serve as a guide to the progress through the cycle:

- Daily Activities
 1. Posting of transactions
 2. Printing a Daily Transaction Report
 3. Maintenance of the inventory files
 4. Printing any On-demand Reports
- Monthly Activities
 1. Do daily activities
 2. Make back-up files
 3. Update history file
 4. Print History Report
 5. Prepare for new month
- Year End Activities
 1. Run final month end activities
 2. Back-up history file
 3. Delete all history records
 4. Prepare for new month

The history records are maintained in a safe place while year end analyses are being performed. Good business practice dictates archiving this information for a period of no less than seven years.

A.3 INTRODUCTION FOR THE OPERATOR

This portion of the guide is directed toward the operator who will be creating the database files, posting transactions and printing the reports.

The Reference Manual provided with your computer explains the procedures for starting the computer, formatting diskettes, copying diskettes, using the text editor, and other general items pertaining to the computer's operation. A good working knowledge of that information is required before attempting to use the Inventory Control System. If necessary, go back and review that material once again.

The inventory system has been set to start right after the computer is turned on. When operator enters the correct password, the main inventory menu is presented. Note that each selection in the main menu contains two reference character. Command mnemonics which you are to type to specify your choice.

A.3.1 Main Menu Selection--MF: Maintain Item Data Base File

Step 1. Type: MF

The program will display the Item File Maintenance Menu and ask you to make a selection.

To Add The New Item To Item Data Base File

Step 2. Type: AD

The program will prompt for item code number.

Step 3. Type: NNNNN <any five digit number>

If that number is already assigned to an item in the database, the program will display a message and wait for the operator to type in an unused number. Otherwise, the program will ask you to enter the description of that item, item name, and so on.

Step 4. After you have filled in all necessary information for that item, the program will ask whether you entered it correctly.

Type: Y <if it is correct>
N <if it is incorrect>

If you answer "Y" the program will add that item to the database and let you go on to next item. If it is "N", the program will disregard that item and let you start over.

Step 5. Repeat steps 3-4 until you have entered all items to the database file.

Step 6. Type: 99999 <in step 3>

When the program prompts for an item code number and you want to exit the add program. The reserved item code '99999' may not be assigned to an item. It is used as an exit flag to the program. In general a numeric value of all 9's is reserved for this purpose.

To Change Information on Existing Items

Step 2. Type: CH

The program will prompt for an item code number.

Step 3. Type: NNNNN <item code number of item
you need to change the info>

If that item is in the database, the program will display all information about that item. Otherwise, the program will display an error message that there is no such item number. You will then enter the correct number or check with your item list report and try again until the program finds the record you want. Note that each line of information contains a reference number (01-23 and 99), for your use in specifying what field you will change.

Step 4. Type: NN <a reference number 01 thru 23>

The program will wait for you to enter the new information for the corresponding field.

Step 5. The program now will ask you to enter another reference number if more change is needed. Repeat step 4 until no more change is needed.

Step 6. Type: 99 <in step 4>

to go on next item and repeat steps 3-5.

Step 7. Type: 99999 <in step 5>

to exit the change program.

To Delete an Existing Item

Step 2. Type: DE

The program will prompt for item code number.

Step 3. Type: NNNNN <the item code number of
item you need to delete
from database>

The program will display the information of that item. Otherwise the message "error" will be displayed. You can now make sure that the information display is for the record you want to delete.

Step 4. Type: Y <if it is correct and you
want to delete>
N <if it is not correct and
you want to skip>

If answer is 'Y' the program will delete that item from the item database file.

Step 5. Repeat steps 3-4 if more items you need to delete.

Type: 99999 <in step 3>

to exit the delete program. The program will ask you to get your printer ready. When the printer is ready, the program will print the list of items you have deleted during the session.

To Query an Existing Items

Step 2. Type: QU

The program will prompt for the item code number.

Step 3. Type: NNNNN <the item code number that you are interested in>

Again, if that item record exists, the program will display all information for that item; otherwise an error message message will be displayed.

Step 4. Repeat step 3 if you need information on more than one item.

Type: 99999 <in step 3>

to exit the query program.

To Exit The Item Database File Maintenance

Step 2. Type : RE

The program will return to the main menu screen. You can then make another selection.

A.3.2 Main Menu Selection--VF: Maintain Vendor File

Step 1. Type: VF

The program will display the Vendor File Maintenance Menu and prompt you to make a selection.

The instructions for adding, changing, deleting or querying the vendor file are similar to those for the item file maintenance above. By now, the operator should be familiar with the instructions for the item file maintenance. If not, the operator can follow the instructions of the item file maintenance for vendor file maintenance.

A.3.3 Main Menu Selection--ET: Enter Transaction

Step 1. Type: ET

The program will display the transaction menu and prompt for selection.

**THIS BOOK
CONTAINS
NUMEROUS PAGES
THAT HAVE INK
SPLOTCHES IN THE
MIDDLE OF THE
TEXT. THIS IS AS
RECEIVED FROM
CUSTOMER.**

**THESE ARE THE
BEST IMAGES
AVAILABLE.**

To Enter A Sale Transaction

Step 2. Type: SA

The program will prompt for an item code number.

Step 3. Type: NNNNN <any five digit number>

If that number does not exist in the item file, the program will display the error message "No Such Item Number: Try Again", until the correct number is entered. The program then displays the name and description of the item. The operator is now prompted for the transaction number, the customer number and the quantity sold. The program then calculates the value of the sale and displays it on the screen.

Step 4. After all necessary information is entered to the system, the program will allow you to check whether you entered it correctly.

Type: Y <if it is correct>
N <if it is incorrect>

If the answer is "Y", the program will update the quantity on hand, period-to-date and year-to-date usage fields in the item file. The transaction record will be added to the transaction file. Otherwise the program will disregard the transaction. The opportunity then is given to enter another transaction.

Step 5. Repeat steps 3-4.

Step 6. Type: 99999 <in step 3>
to exit this program.

To Enter An Adjustment to Quantity-on-Hand Transaction

Step 2. Type: AD

The program will prompt for item code number.

Step 3. Type: NNNNN <any five digit numbers>

If that number does not exist in the Item file, the program will display an error message "No Such Item Number: Try Again", until the correct number is entered. The program then displays the name and description of the item. The operator is now prompted for the transaction number, the operator id, the adjustment amount and the reason for adjustment. The program then calculates the total amount and displays it on the screen.

Step 4. The program will prompt for correction of the transaction.

Type: Y <if it is correct>
 N <if it is incorrect>

If the answer is "Y", the program will update the quantity on hand field in the Item file and the transaction record will be added to the transaction file. Otherwise the program will disregard the transaction. The opportunity is given to enter another transaction.

Step 5. Repeat step 3-4.

Step 6. Type: 99999 <in step 3>

To exit this program.

To Post An Order To Vendor

Step 2. Type: PO

The program will prompt for an item code number.

Step 3. Type: NNNNN <any five digit numbers>

The program will check for the item code validation. If it is a valid code, the program will display the name and description of the item. The operator is now prompted for the transaction number, vendor code, the quantity order.

Step 4. The program will prompt for correction of the transaction.

Type: Y <if it is correct>
N <if it is incorrect>

If the answer is "Y", the program will update the quantity on order, the date of order fields in the Item file.

Step 5. Repeat step 3-4.

Step 6. Type: 99999 <in step 3>

To exit this program.

To Enter A Receipt Item From Vendor

Step 2. Type: RI

The program will prompt for item code number.

Step 3. Type: NNNNN <any five digit numbers>

The program will check for the validity of item code. If it is a valid code, the program will display the name and description of the item. The operator is now prompted for the transaction number, vendor code and the received quantity.

Step 4. The program will prompt for correction of the transaction.

Type: Y <if it is correct>
 N <if it is incorrect>

If the answer is "Y", the program will update the quantity on hand, the date of last receipt, the order date and the quantity ordered fields in the Item file.

Step 5. Repeat step 3-4.

Step 6. Type: 99999 <in step 3>

To exit this program.

To Print The Daily Transaction Report

Step 2. Type: PP

The program will give an opportunity to start the printer and print the report. This selection should be done when there are no more transactions to be processed. Otherwise the transactions that are entered to the system after printing the report will appear in the next day's report. When the program finished, the operator has an opportunity to make other selection or return to Main Menu.

To Return To The Main Menu Screen

Step 2. Type: RE

The program will exit the current menu and display the Main Selection Menu.

A.3.4 Main Menu Selection--EP: End of Period Update & Report

Step 1. Type: EP

The program will display the End of Period Maintenance Menu and prompt you to make a selection.

To Update the History File

Important Note: Before doing the following steps, be sure that all postings for the current month have been completed. A month end copy of the data diskette should be made and retained for back-up and audit purposes.

Step 2. Type: UH

The program will response:

*****Updating History File In Progress*****
Program will take a few minutes to completed
*****Do not Interrupt While in Progress*****

Once the program finished, the opportunity is given to make another selection.

To Print the History Report

Step 2. Type: HR

The program will respond:

```
Please enter the customer id or press return  
** Please set Printer to Top of the Form **  
Program will take a few minutes to completed  
*****Do not Interrupt While in Progress*****
```

The opportunity is given to print a selected customer or all customers. Pressing return will cause the program to print the report for all customers otherwise the operator needs to type the customer id of the selected customer. When the printer is ready, the report is automatically printed. Once the program finished, the opportunity is given to make another selection.

A.3.5 Main Menu Selection--RP: Inventory Report

Step 1. Type: RP

The program will display the Report Menu and prompt you to make a selection.

Step 2. Type: IC <for Item Catalog>
or: LV <for List of Vendors>
or: PI <for Physical Inventory Worksheet>
or: EX <for List of Expired Items>
or: RA <for Reorder Advice Report>
or: RE <to return to Main Menu>

In all reports, the operator is given an opportunity to start the printer and align the paper before the printing of the report is started. Reports in this menu can be obtained repeatedly without altering the file contents. When the report is printed, the operator can make another selection.

A.3.6 Main Menu Selection--BK: Back-Up Inventory Files

Step 1. Type: BK

The program will display the Back-up Files Menu and prompt you to make a selection.

Step 2. Type: IF <to copy Item file>
or: HF <to copy History file>
or: VF <to copy Vendor file>
or: CF <to copy Customer file>
or: TF <to copy Transaction file>
or: RE <to return to Main Menu>

This program provides the facility to copy Inventory files. The transaction file should be copied everyday. The Item file, the History file, the Customer file, and the Vendor file should be copied at least once a month before month end processing starts. The operator selects the file which needs to be copied in step 2. The program will prompt the operator to insert the back-up diskette in the diskette drive. Once the diskette drive is ready, the program will copy the file including the index file to the diskette disk. After the program finished copying the file, the operator can make another selection or return to the Main Menu.

A.3.7 Main Menu Selection--EN: End System Selected

Step 1. Type: EN

The final selection provides means to exit from the Inventory System. If the full dBASE II package is used, the control will be returned to the dBASE environment. To exit the dBASE environment, typing the dBASE command "QUIT" will cause control to return to the microcomputer operating system. If the dBASE RUNTIME is used, the control will be return to the operating system environment upon exit from the Inventory System program.

Appendix B

SAMPLE SYSTEM REPORTS

The sample reports included below were prepared using the Inventory Control System.

- Daily Transaction Report
- History Report
- List of Master Item
- List of Master Vendor
- Physical Inventory Worksheet
- List of Expired Item
- Reorder Advice Report

COLLEGE OF VETERINARY MEDICINE
DEPARTMENT OF SURGERY AND MEDICINE
REPORT OF SALE TRANSACTIONS MADE ON: 06/06/84

ITEM-NO	ITEM NAME	DESCRIPTION	SALE-TO	TRAN-NO	QUANTITY	PRICE	TOTAL-AMT
10311	ACD BOTTLE	250ML LIQUID PER BTL	KABSU	000001	3	\$55.14	\$15.42
11211	ACEFROM. MELEATE TABS	10MG 100 TABS PER BTL	KABSU	000002	2	\$12.09	\$24.18
13010	A-H INJECTION	11.36MG/ML LIQ PER ML	KABSU	000003	10	\$80.06	\$80.60
14401	ADENOSINE	LIQ PER ML	KABSU	000051	2	\$15.00	\$30.00
15021	A.S.A. COMPOUND	100 TABS PER BTL	KABSU	192900	5	\$33.33	\$16.65
71911	ADAMS FLEA OFF MIST	4 OZ LIQ PER BTL	KABSU	282829	10	\$4.12	\$41.20

COLLEGE OF VETERINARY MEDICINE
DEPARTMENT OF SURGERY AND MEDICINE
REPORT OF RECEIVED ITEMS FROM VENDORS MADE ON: 06/06/84

ITEM-NO	ITEM NAME	DESCRIPTION	REC-FR	TRAN-NO	QUANTITY
10311	ACD BOTTLE	250ML LIQUID PER BTL	00002	102010	10
10611	AA 1000	500ML LIQUID PER BTL	00003	102011	13

COLLEGE OF VETERINARY MEDICINE
DEPARTMENT OF SURGERY AND MEDICINE
REPORT OF POST ORDERS TO VENDORS MADE ON: 06/06/84

ITEM-NO	ITEM NAME	DESCRIPTION	POST-TO	TRAN-NO	QUANTITY
10611	AA 1000	500ML LIQUID PER BTL	00001	192910	10
11211	ACEFROM. MELEATE TABS	10MG 100 TABS PER BTL	00002	102011	10
11221	ACEFROMAZINE	10MG PER TAB	00003	388299	10
13010	A-H INJECTION	11.36MG/ML LIQ PER ML	00001	100002	10

COLLEGE OF VETERINARY MEDICINE
DEPARTMENT OF SURGERY AND MEDICINE
REPORT OF ADJUSTMENT TRANSACTIONS MADE ON: 06/06/84

ITEM-NO	ITEM NAME	DESCRIPTION	TRAN-NO	QUANTITY	PRICE	TOTAL-AMT	REMARKS
10311	ACD BOTTLE	250ML LIQUID PER BTL	000011	10	\$5.14	\$51.40	RETURN
14401	ADENOSINE	LIQ PER ML	123001	1	\$15.00	\$15.00	UNUSED ABLE
71911	ADAMS FLEA OFF MIST	4 OZ LIQ PER BTL	727289	5	\$4.12	\$20.60	EXPIRED

SUMMARY OF TRANSACTION MADE ON: 06/06/84

TOTAL NUMBER OF TRANSACTIONS. . . .	15
TOTAL SALES.\$128.05
TOTAL ADJUSTMENTS.\$87.00

COLLEGE OF VETERINARY MEDICINE
 DEPARTMENT OF SURGERY AND MEDICINE
 HISTORY REPORT BY CUSTOMER: ANATOMY & PHYSIOLOGY - RESEARCH
 FOR THE PERIOD ENDING: 06/06/84

ITEM-NO	GENERIC NAME	DESCRIPTION	PTD-USAGE	YTD-USAGE	TOTAL-PTD	TOTAL-YTD
11211	ACEPROMAZINE MALEATE TABLETS	10MG 100 TABS PER BTL	5	10	\$960.45	\$9120.90
13010	DOXYLAMINE SUCCINATE	11.36MG/ML LIQ PER ML	10	20	\$960.60	\$9991.20
15021	ASPIRIN-PHENACETIN-CAFFEINE	100 TABS PER BTL	20	40	\$966.60	\$9133.20
71911	PYRETHRINS .15%	4 OZ LIQ PER BTL	5	10	\$920.60	\$9941.20
GRAND TOTAL:					\$9148.25	\$99296.50

COLLEGE OF VETERINARY MEDICINE
DEPARTMENT OF SURGERY AND MEDICINE
HISTORY REPORT BY CUSTOMER: KS. ARTIFICIAL BREEDING SERVICE UNIT
FOR THE PERIOD ENDING: 06/06/84

ITEM-NO	GENERIC NAME	DESCRIPTION	PTD-USAGE	YTD-USAGE	TOTAL-PTD	TOTAL-YTD
10311	BLOOD COLLECTION BOTTLE	250ML LIQUID PER BTL	6	6	\$330.84	\$330.84
11211	ACEPROMAZINE MALEATE TABLETS	10MG 100 TABS PER BTL	4	4	\$48.36	\$48.36
13010	DOXYLAMINE SUCCINATE	11.36MG/ML LIQ PER ML	20	20	\$31.20	\$31.20
14401	ADENOSINE #5	LIQ PER ML	4	4	\$60.00	\$60.00
15021	ASPIRIN-PHENACETIN-CAFFEINE	100 TABS PER BTL	10	10	\$33.30	\$33.30
71911	PYRETHRINS .15Z	4 OZ LIQ PER BTL	20	20	\$82.40	\$82.40
GRAND TOTAL:					\$256.10	\$256.10

COLLEGE OF VETERINARY MEDICINE
DEPARTMENT OF SURGERY AND MEDICINE
LIST OF ITEMS IN THE INVENTORY
DATE PRINTED: 06/06/84

ITEM-NO	DESCRIPTION	LOCATION	ON-HAND	UNIT-COST
13010	A-H INJECTION 11.36MG/ML LIQ PER ML DOXYLAMIE SUCCINATE	M2AT	30	\$90.06
15021	A.S.A. COMPOUND 100 TABS PER BTL ASPIRIN-PHENACETIN-CAFFEINE	M8B1	19	\$93.33
10611	AA 1000 500ML LIQUID PER BTL AA 1000	M8A5	36	\$18.00
10311	ACD BOTTLE 250ML LIQUID PER BTL BLOOD COLLECTION BOTTLE	M8A5	47	\$95.14
10231	ACE BANDAGE 4 INCH ROLL ELASTIC BANDAGE	M7A3	10	\$93.00
11211	ACEFROM. MELEATE TABS 10MG 100 TABS PER BTL ACEFROMAZINE MALEATE TABLETS	M8B2	26	\$12.09
11221	ACEFROMAZINE 10MG PER TAB ACEFROMAZINE	M8B2	10	\$90.33
71911	ADAMS FLEA OFF MIST 4 OZ LIQ PER BTL PYRETHRINS .15%	M3A7	40	\$94.12
14401	ADENOSINE LIQ PER ML ADENOSINE #5	M7B5	25	\$15.00

COLLEGE OF VETERINARY MEDICINE
DEPARTMENT OF SURGERY AND MEDICINE
LIST OF VENDORS IN THE INVENTORY SYSTEM
DATE PRINTED: 06/06/84

VENDOR-NO	NAME & ADDRESS	PHONE	PURCHASE-ID	CONTACT
00001	BERRY'S DRUG CENTER 409 FOYNTZ MANHATTAN	913-776-8833		
			NY 66502	
00002	DUNNE'S PHARMACY 2429 CLAFLIN ROAD MANHATTAN	913-539-2345		
			KS 66502	
00003	KELLSTROM PHARMACY 1860 CLAFLIN ROAD MANHATTAN	913-776-1200		
			KS 66502	

COLLEGE OF VETERINARY MEDICINE
DEPARTMENT OF SURGERY AND MEDICINE
PHYSICAL INVENTORY WORKSHEET
DATE PRINTED: 06/06/84

ITEM-NO	DESCRIPTION	LOCATION	ON-HAND	ACTUAL-ON-HAND
13010	A-H INJECTION 11.36MG/ML LIQ PER ML DOXYLAMINE SUCCINATE	M2AT	30	
15021	A.S.A. COMPOUND 100 TABS PER BTL ASPIRIN-PHENACETIN-CAFFEINE	M8B1	19	
10611	AA 1000 500ML LIQUID PER BTL AA 1000	M8A5	36	
10311	ACD BOTTLE 250ML LIQUID PER BTL BLOOD COLLECTION BOTTLE	M8A5	47	
10231	ACE BANDAGE 4 INCH ROLL ELASTIC BANDAGE	M7A3	10	
11211	ACEFROM. MELEATE TABS 10MG 100 TABS PER BTL ACEFROMAZINE MALEATE TABLETS	M8B2	26	
11221	ACEFROMAZINE 10MG PER TAB ACEFROMAZINE	M8B2	10	
71911	ADAMS FLEA OFF MIST 4 OZ LIQ PER BTL PYRETHRINS .15%	M3A7	40	
14401	ADENOSINE LIQ PER ML ADENOSINE #5	M7B5	25	
15051	ADRIAMYCIN 10MG PER VIAL DOXOROBICIN HCL	M1C3	41	

COLLEGE OF VETERINARY MEDICINE
DEPARTMENT OF SURGERY AND MEDICINE
LIST OF EXPIRED ITEM IN THE INVENTORY SYSTEM
DATE PRINTED: 06/06/84

ITEM-NO	DESCRIPTION	LOCATION	ON-HAND	EXPIRED-DATE
10311	ACD BOTTLE 250ML LIQUID PER BTL BLOOD COLLECTION BOTTLE	M8A5	37	01/84
10611	AA 1000 500ML LIQUID PER BTL AA 1000	M8A5	23	02/84

COLLEGE OF VETERINARY MEDICINE
 DEPARTMENT OF SURGERY AND MEDICINE
 REORDER ADVISE REPORT FROM ITEM DATA BASE
 DATE PRINTED: 06/06/84

ITEM-NO	DESCRIPTION	ONHAND	REORDER-LVL	REORDER-QTY	VENDOR-ID
10611	AA 1000 500ML LIQUID PER BTL AA 1000	23	25	10	11111 22222 33333
11211	ACEPRON. MELEATE TABS 10MG 100 TABS PER BTL ACEPRONAZINE MALEATE TABLETS	26	35	15	00001 00002 00003
11221	ACEPRONAZINE 10MG PER TAB ACEPRONAZINE	10	10	10	00005
13010	A-H INJECTION 11.36MG/ML LIQ PER ML DOXYLAMIE SUCCINATE	30	30	10	00008

Appendix C
SYSTEM SOURCE CODE

```

****THIS PROGRAM PROVIDE A MAIN MENU FOR INVENTORY SYSTEM
*INITIALIZE
SET TALK OFF
ERASE
IF DATE() = '00/00/00'
  @ 7,10 SAY 'Provide Date as MM/DD/YY' GET tdate
  READ
ELSE
  STORE DATE() TO tdate
ENDIF
SET ESCAPE ON
SET COLON OFF
ERASE
STORE ' ' TO processing
*PRESENT MAIN SCREEN TITLES
  @ 7,22 SAY 'Inventory Control Package Version 1.0'
  @ 9,32 SAY 'Today is '+tdate
  @ 11,23 SAY 'Copyright 1984 KSU Computing Center'
  @ 13,25 SAY 'Press Any Key To Run Inventory' GET processing
  READ
ERASE
DO invmenu
STORE T TO processing
DO WHILE processing
  *GET USERS SELECTION
  STORE ' ' TO replyy
  @ 21,53 GET replyy
  READ
  *INVOKE DESIRED SYSTEM COMPONENT
  DO CASE
    CASE !(replyy) = 'MF'
      DO invitem
      LOOP
    CASE !(replyy) = 'VF'
      DO invendor
      LOOP
    CASE !(replyy) = 'ET'
      DO invet
      LOOP
    CASE !(replyy) = 'EP'
      DO inveop
      LOOP
    CASE !(replyy) = 'BK'
      DO invback
      LOOP
    CASE !(replyy) = 'RP'
      DO invrept
      LOOP
    CASE !(replyy) = 'EN'
      STORE F TO processing
  ENDCASE
ENDDO processing
RELEASE replyy, processing
*CLEANUP AND QUIT

```

CANCEL

*****END INVITEM.PRG*****

****THIS PROGRAM PROVIDES A MENU FOR MAINTANANCE ITEM FILE

STORE T TO updating

DO WHILE updating

*PRESENT A SCREEN OF CHOICES

DO itemmenu

*PROMPT, ACCEPT AN ANSWER

STORE ' ' TO ans

@ 7,50 GET ans

READ

*BRANCH TO APPROPRIATE SUPPORT ROUTINE

DO CASE

CASE !(ans)='AD'

DO invadd

LOOP

CASE !(ans)='DE'

DO invdet

LOOP

CASE !(ans)='CH'

DO invchg

LOOP

CASE !(ans)='QU'

DO invqu

LOOP

CASE !(ans)='RE'

STORE F TO updating

ENDCASE

ENDDO updating

*CLEAN UP AND QUIT

RELEASE updating, ans

ERASE

*PRESENT MAINMENU

DO invmenu

RETURN

*****END INVITEM.PRG*****

```

****THIS PROGRAM ADDS A NEW ITEM TO THE ITEM FILE
USE invitem INDEX invitem
DO inititem
STORE T TO continue
DO WHILE continue
  STORE ' ' TO goner
  @ 9,14 SAY ' ' GET goner
  @ 22,16 SAY ' '
  @ 23,0 SAY ' '
  READ
  IF goner = '99999'
    DO inititem
    STORE F TO continue
  LOOP
ENDIF
SET EXACT ON
FIND &goner
IF # = 0
  CLEAR GET
  *PROMPT NEW INFORMATION
  @ 9,40 GET mitem:desc
  @ 10,14 GET mlocation
  @ 10,40 GET mitem:name
  @ 11,14 GET mdrug:code
  @ 11,40 GET mgeneric
  *PROMPT COSTS
  @ 12,14 GET mact:cost
  @ 12,55 GET mnor:cost
  READ
  STORE mnor:cost * .5 + mnor:cost TO mret:cost
  @ 13,15 SAY mret:cost USING '999.99'
  @ 13,55 GET mq:o:h
  *PROMPT REORDER INFO
  @ 14,15 GET mreord:pt PICTURE '999'
  @ 15,17 GET mreord:amt PICTURE '999'
  @ 15,55 GET mdate:ord PICTURE '99/99/99'
  @ 16,17 GET mdlastrec PICTURE '99/99/99'
  @ 16,59 GET mvitem:no
  @ 17,17 GET mdexpire
  @ 17,59 GET mnat:code PICTURE '99/99'
  @ 18,18 GET mbuy:code
  @ 18,55 GET mst:code
  *PROMPT VENDORS
  @ 19,17 GET mvendor1
  @ 19,55 GET mvendor2
  @ 20,17 GET mvendor3
  READ
  @ 22,16 SAY 'Is This Correct? (Y/N): '
  STORE ' ' TO answer
  @ 22,40 SAY ' ' GET answer
  READ
  @ 22,41 SAY ' '
  IF !(answer) = 'Y'
    APPEND BLANK

```

```

*PUT FIELDS IN RECORD
REPLACE item:code WITH goner
REPLACE location WITH mlocation
*REPLACE DESCRIPTIONS
  REPLACE item:name WITH mitem:name
  REPLACE item:desc WITH mitem:desc
  REPLACE generic WITH mgeneric
REPLACE drug:code WITH mdrug:code
REPLACE disc:code WITH mdisc:code
*REPLACE COSTS
  REPLACE act:cost WITH mact:cost
  REPLACE ret:cost WITH mret:cost
  REPLACE nor:cost WITH mnor:cost
REPLACE q:o:h WITH mq:o:h
*REPLACE REORDER INFO
  REPLACE reord:pt WITH mreord:pt
  REPLACE reord:amt WITH mreord:amt
  REPLACE date:ord WITH $(mdate:ord,7,2) +$(mdate:ord,1,2
  +$(mdate:ord,4,2)
  REPLACE dlastrec WITH $(mdlastrec,7,2) +$(mdlastrec,1,2
  +$(mdlastrec,4,2)
  REPLACE dexpire WITH $(mdexpire,4,2) +$(mdexpire,1,2)
  REPLACE nat:code WITH mnat:code
  REPLACE buy:code WITH mbuy:code
  REPLACE st:code WITH mst:code
*REPLACE VENDORS
  REPLACE vitem:no WITH mvitem:no
  REPLACE vendor1 WITH mvendor1
  REPLACE vendor2 WITH mvendor2
  REPLACE vendor3 WITH mvendor3
DO inititem
ENDIF
ELSE
  @ 22,0 SAY chr(7)
  @ 22,16 SAY 'Duplicate Item Code: TRY AGAIN'
  WAIT
ENDIF
ENDDO continue
RELEASE continue, answer, goner
RELEASE ALL LIKE m*
RETURN
*****END INVADD.PRG*****

```

```

****THIS PROGRAM DELETES AN EXISTING ITEM FROM THE ITEM FILE
USE invitem INDEX invitem
SELECT SECONDARY
USE olditem
SELECT PRIMARY
STORE T TO deletes
DO WHILE deletes
  STORE ' ' TO goner
  @ 9,14 SAY ' ' GET goner
  @ 22,16 SAY '
  @ 23,0 SAY '
  READ
  IF goner = '99999'
    STORE F TO deletes
    LOOP
  ENDIF
  FIND &goner
  IF # = 0
    @ 22,0 SAY CHR(7)
    @ 22,16 SAY 'No Such Item Code: Try Again'
    WAIT
    LOOP
  ENDIF
  *SHOW RECORD CONTENTS
  @ 9,40 SAY item:desc
  @ 10,14 SAY location
  @ 10,40 SAY item:name
  @ 11,14 SAY drug:code
  @ 11,40 SAY generic
  *PROMPT COSTS
  @ 12,14 SAY act:cost
  @ 12,55 SAY nor:cost
  @ 13,15 SAY ret:cost
  @ 13,55 SAY q:o:h
  *PROMPT REORDER INFO
  @ 14,15 SAY reord:pt
  @ 15,17 SAY reord:amt
  @ 15,55 SAY date:ord
  @ 16,17 SAY dlastrec
  @ 17,17 SAY dexpire
  @ 17,59 SAY nat:code
  @ 18,18 SAY buy:code
  @ 18,55 SAY st:code
  *PROMPT VENDORS
  @ 16,59 SAY vitem:no
  @ 19,17 SAY vendor1
  @ 19,55 SAY vendor2
  @ 20,17 SAY vendor3
  *PROMPT, CONFIRM DELETION
  STORE ' ' TO answer
  @ 22,16 SAY 'Is This The Item To Be Deleted? (Y/N):'
  @ 22,55 SAY ' ' GET answer
  READ
  IF !(answer) = 'Y'

```

```

*ADD RECORD TO OLDITEM
SELECT SECONDARY
APPEND BLANK
REPLACE item:code WITH goner
REPLACE location WITH p.location
*REPLACE DESCRIPTIONS
REPLACE item:name WITH p.item:name
REPLACE item:desc WITH p.item:desc
REPLACE generic WITH p.generic
REPLACE drug:code WITH p.drug:code
REPLACE disc:code WITH p.disc:code
*REPLACE COSTS
REPLACE act:cost WITH p.act:cost
REPLACE ret:cost WITH p.ret:cost
REPLACE nor:cost WITH p.nor:cost
REPLACE q:o:h WITH p.q:o:h
*REPLACE REORDER INFO
REPLACE reord:pt WITH p.reord:pt
REPLACE reord:amt WITH p.reord:amt
REPLACE date:ord WITH p.date:ord
REPLACE dlastrec WITH p.dlastrec
REPLACE dexpire WITH p.dexpire
REPLACE nat:code WITH p.nat:code
REPLACE buy:code WITH p.buy:code
REPLACE st:code WITH p.st:code
*REPLACE VENDORS
REPLACE vendor1 WITH p.vendor1
REPLACE vendor2 WITH p.vendor2
REPLACE vendor3 WITH p.vendor3
SELECT PRIMARY
STORE STR(5) TO number
DELETE RECORD &number
ENDIF
ENDDO deletes
@ 22,0 SAY CHR(7)
@ 22,4 SAY '>>>>>' One Moment Please <<<<<
@ 22,33 SAY '
SELECT PRIMARY
PACK
*PRINT THE DELETED RECORDS
SELECT SECONDARY
USE OLDITEM
GO TOP
@ 22,4 SAY '>>>>>' Please Set Printer To Top Of Form <<<<<
WAIT
*SET PRINTER
SET FORMAT TO PRINT
SET CONSOLE OFF
SET HEADING TO DEPARTMENT OF SURGERY AND MEDICINE
REPORT FORM invdet TO PRINT
EJECT
SET CONSOLE ON
SET FORMAT TO SCREEN
USE olditem

```



```
DELETE ALL
PACK
RELEASE deletes, answer, reply, goner, number
RELEASE ans, i
RETURN
*****END INVDET.PRG*****
```

```

*****THIS PROGRAM MODIFIES THE INFORMATION ON THE ITEM FILE
USE invitem INDEX invitem
STORE T TO changes
DO WHILE changes
  STORE ' ' TO goner
  @ 9,14 SAY ' ' GET goner
  @ 22,16 SAY ' '
  @ 23,0 SAY ' '
  @ 23,16 SAY ' '
  READ
  IF goner = '99999'
    STORE F TO changes
    LOOP
  ENDIF
  FIND &goner
  IF # = 0
    @ 22,0 SAY CHR(7)
    @ 22,16 SAY 'No Such Item Code: Try Again'
    WAIT
    LOOP
  ENDIF
  *SHOW RECORD CONTENTS
  @ 9,40 SAY item:desc
  @ 10,14 SAY location
  @ 10,40 SAY item:name
  @ 11,14 SAY drug:code
  @ 11,40 SAY generic
  *PROMPT COSTS
  @ 12,14 SAY act:cost USING '999.99'
  @ 12,55 SAY nor:cost USING '999.99'
  @ 13,15 SAY ret:cost USING '999.99'
  @ 13,55 SAY q:o:h
  *PROMPT REORDER INFO
  @ 14,15 SAY reord:pt USING '999'
  @ 15,17 SAY reord:amt USING '999'
  STORE $(date:ord,3,2)+'/'+'$(date:ord,5,2)+'/'+'$(date:ord,1
  TO tdate:ord
  @ 15,55 SAY tdate:ord
  STORE $(dlastrec,3,2)+'/'+'$(dlastrec,5,2)+'/'+'$(dlastrec,1
  TO tdlastrec
  @ 16,17 SAY tdlastrec
  @ 16,59 SAY vitem:no
  STORE $(dexpire,3,2)+'/'+'$(dexpire,1,2) TO texpire
  @ 17,17 SAY texpire
  @ 17,59 SAY nat:code
  @ 18,18 SAY buy:code
  @ 18,55 SAY st:code
  *PROMPT VENDORS
  @ 19,17 SAY vendor1
  @ 19,55 SAY vendor2
  @ 20,17 SAY vendor3
  *PROMPT, CONFIRM TO CHANGE
  STORE ' ' TO answer
  @ 22,16 SAY 'Is This The Correct Record? (Y/N):'

```

```

@ 22,50 GET answer
READ
IF !(answer) = 'Y'
  CLEAR GET
  STORE T TO continue
  DO WHILE continue
    STORE ' ' TO number
    @ 23,16 SAY 'Enter The Number Of Selection: '
    @ 23,49 GET number
    READ
    DO CASE
      CASE number = '02'
        STORE ' ' TO temp
        @ 9,40 GET temp
        READ
        REPLACE item:desc WITH temp
      CASE number = '03'
        STORE ' ' TO temp
        @ 10,14 GET temp
        READ
        REPLACE location WITH temp
      CASE number = '04'
        STORE ' ' TO temp
        @ 10,40 GET temp
        READ
        REPLACE item:name WITH temp
      CASE number = '05'
        STORE ' ' TO temp
        @ 11,14 GET temp
        READ
        REPLACE drug:code WITH temp
      CASE number = '06'
        STORE ' ' TO temp
        @ 11,40 GET temp
        READ
        REPLACE gereneric WITH temp
      *DUMMY EXTENDER
      CASE number = '07'
        STORE 000.00 TO temp
        @ 12,14 GET temp PICTURE '999.99'
        READ
        REPLACE act:cost WITH temp
      CASE number = '08'
        *DUMMY 08 EXTENDER
        STORE 000.00 TO temp
        @ 12,55 GET temp PICTURE '999.99'
        READ
        REPLACE nor:cost WITH temp
      CASE number = '09'
        STORE 000.00 TO temp
        @ 13,15 GET temp PICTURE '999.99'
        READ
        REPLACE ret:cost WITH temp
      CASE number = '11'

```

```

*DUMMY 11 EXTENDER
  STORE 000 TO temp
  @ 14,15 GET temp      PICTURE '999'
  READ
  REPLACE reord:pt      WITH temp
CASE number = '13'
  STORE 000 TO temp
  @ 15,17 GET temp      PICTURE '999'
  READ
  REPLACE reord:amt     WITH temp
*ANOTHER DUMMY
*DUMMY FOR 14-17
  CASE number = '14'
    STORE ' ' TO temp
    @ 15,55 GET temp     PICTURE 'mm/dd/yy'
    READ
    REPLACE date:ord     WITH $(temp,7,2)+$(temp,1,2)+$(temp
CASE number = '15'
    STORE ' ' TO temp
    @ 16,17 GET temp     PICTURE 'mm/dd/yy'
    READ
    REPLACE dlastrec     WITH $(temp,7,2)+$(temp,1,2)+$(temp
CASE number = '16'
    STORE ' ' TO temp
    @ 16,59 GET temp
    READ
    REPLACE vitem:no     WITH temp
CASE number = '17'
    STORE ' ' TO temp
    @ 17,17 GET temp     PICTURE 'mm/yy'
    READ
    REPLACE dexpire     WITH $(temp,4,2)+$(temp,1,2)
CASE number = '18'
    STORE ' ' TO temp
    @ 17,59 GET temp
    READ
    REPLACE nat:code     WITH temp
CASE number = '19'
*DUMMY 19 EXTENDER
  STORE ' ' TO temp
  @ 18,18 GET temp
  READ
  REPLACE buy:code      WITH temp
CASE number = '20'
  STORE ' ' TO temp
  @ 18,55 GET temp
  READ
  REPLACE st:code       WITH temp
LAST DUMMY
CASE number = '21'
  STORE ' ' TO temp
  @ 19,17 GET temp
  READ
  REPLACE vendor1       WITH temp

```

```
CASE number = '22'
  STORE '      ' TO temp
  @ 19,55 GET temp
  READ
  REPLACE vendor2 WITH temp
CASE number = '23'
  STORE '      ' TO temp
  @ 20,17 GET temp
  READ
  REPLACE vendor3 WITH temp
CASE number = '99'
  STORE F TO continue
ENDCASE
ENDDO continue
ENDIF
ENDDO changes
RELEASE changes, goner, continue, answer, number
RELEASE ALL LIKE t*
RETURN
*****END INVCHG.PRG*****
```

```

*****THIS PROGRAM PROVIDES THE INFORMATION OF EXISTING ITEM
USE invitem INDEX invitem
STORE T TO queries
STORE ' ' TO goner
DO WHILE queries
  @ 9,14 SAY ' ' GET goner
  *CLEAR LINES
  @ 22,16 SAY '
  @ 23,0 SAY '
  READ
  IF goner = '99999'
    STORE F TO queries
    LOOP
  ENDIF
  FIND &goner
  IF # = 0
    @ 22,0 SAY CHR(7)
    @ 22,16 SAY 'No Such Item Code: Try Again'
    WAIT
    LOOP
  ENDIF
  *SHOW RECORD CONTENTS
  @ 9,40 SAY item:desc
  @ 10,14 SAY location
  @ 10,40 SAY item:name
  @ 11,14 SAY drug:code
  @ 11,40 SAY generic
  *PROMPT COSTS
  @ 12,14 SAY act:cost
  @ 12,55 SAY nor:cost
  @ 13,15 SAY ret:cost
  @ 13,55 SAY q:o:h
  *PROMPT REORDER INFO
  @ 14,15 SAY reord:pt
  @ 15,17 SAY reord:amt
  @ 15,55 SAY date:ord
  @ 16,17 SAY dlastrec
  @ 16,59 SAY vitem:no
  @ 17,17 SAY dexpire
  @ 17,59 SAY nat:code
  @ 18,18 SAY buy:code
  @ 18,55 SAY st:code
  *PROMPT VENDORS
  @ 19,17 SAY vendor1
  @ 19,55 SAY vendor2
  @ 20,17 SAY vendor3
ENDDO queries
RELEASE queries, goner
RETURN
*****END INVQU.PRG*****

```

```

****THIS PROGRAM PROVIDES A MENU TO MAINTAIN THE VENDOR FILE
STORE T TO updating
DO WHILE updating
  *PRESENT A SCREEN OF CHOICES
  DO venmenu
  *PROMPT, ACCEPT AN ANSWER
  STORE ' ' TO ans
  @ 7,50 GET ans
  READ
  *BRANCH TO APPROPRIATE SUPPORT ROUTINE
  DO CASE
    CASE !(ans)='AD'
      DO venadd
      LOOP
    CASE !(ans)='DE'
      DO vendet
      LOOP
    CASE !(ans)='CH'
      DO venchg
      LOOP
    CASE !(ans)='QU'
      DO venqu
      LOOP
    CASE !(ans)='RE'
      STORE F TO updating
  ENDCASE
ENDDO updating
*CLEAN UP AND QUIT
RELEASE updating, ans
ERASE
*PRESENT MAINMENU
DO invmenu
RETURN
*****END INVENDOR.PRG*****

```

```

****THIS PROGRAM ADDS A NEW VENDOR TO THE VENDOR FILE
USE invendor INDEX invendor
STORE T TO continue
DO WHILE continue
  *INIT VARIABLES
  STORE ' ' TO goner
  STORE ' ' TO mname
  STORE ' ' TO maddress
  STORE ' ' TO mcity
  STORE ' ' TO mstate
  STORE ' ' TO mzipcode
  STORE ' ' TO mphone
  STORE ' ' TO mcontact
  STORE ' ' TO mpur:id
  @ 10,17 GET goner
  @ 20,16 SAY ' '
  @ 22,0 SAY ' '
  READ
  IF goner = '99999'
    DO inititem
    STORE F TO continue
    LOOP
  ENDIF
  SET EXACT ON
  FIND &goner
  IF # = 0
    CLEAR GET
    *PROMPT NEW INFORMATION
    @ 11,17 GET mname
    @ 12,17 GET maddress
    @ 13,17 GET mcity
    @ 14,17 GET mstate
    @ 15,17 GET mzipcode
    @ 16,17 GET mphone PICTURE '999-999-9999'
    @ 17,17 GET mcontact
    @ 18,17 GET mpur:id
  READ
  @ 20,20 SAY 'Is This Correct? (Y/N): '
  STORE ' ' TO answer
  @ 20,48 SAY ' ' GET answer
  READ
  @ 20,49 SAY ' '
  IF !(answer) = 'Y'
    APPEND BLANK
    *PUT FIELDS IN RECORD
    REPLACE vendor:id WITH goner
    REPLACE name WITH mname
    REPLACE address WITH maddress
    REPLACE city WITH mcity
    REPLACE state WITH mstate
    REPLACE zipcode WITH mzipcode
    REPLACE phone WITH mphone
    REPLACE contact WITH mcontact
    REPLACE pur:id WITH mpur:id

```



```
ENDIF
ELSE
  @ 20,0 SAY chr(7)
  @ 20,16 SAY 'Duplicate Vendor Code: TRY AGAIN'
  WAIT
ENDIF
ENDDO continue
REINDEX
RELEASE continue, answer, goner
RELEASE ALL LIKE m*
RETURN
*****END VENADD.PRG*****
```

```

****THIS PROGRAM DELETES THE EXISTING VENDOR FROM THE VENDOR F
USE invendor INDEX invendor
SELECT SECONDARY
USE ovendor
SELECT PRIMARY
STORE T TO deletes
DO WHILE deletes
  STORE ' ' TO goner
  @ 10,17 GET goner
  *CLEAR LINES
  @ 22,16 SAY '
  @ 23,0 SAY '
  @ 23,16 SAY '
  READ
  IF goner = '99999'
    STORE F TO deletes
    LOOP
  ENDIF
  FIND &goner
  IF # = 0
    @ 22,0 SAY CHR(7)
    @ 22,16 SAY 'No Such Item Code: Try Again'
    WAIT
    LOOP
  ENDIF
  *SHOW RECORD CONTENTS
  @ 11,17 SAY name
  @ 12,17 SAY address
  @ 13,17 SAY city
  @ 14,17 SAY state
  @ 15,17 SAY zipcode
  @ 16,17 SAY phone
  @ 17,17 SAY contact
  @ 18,17 SAY pur:id
  *PROMPT, CONFIRM DELETION
  STORE ' ' TO answer
  @ 23,16 SAY 'Is This The Vendor To Be Deleted? (Y/N):'
  @ 23,57 SAY ' ' GET answer
  READ
  IF !(answer) = 'Y'
    *ADD RECORD TO OLDITEM
    SELECT SECONDARY
    APPEND BLANK
    REPLACE vendor:id WITH goner
    REPLACE name WITH p.name
    REPLACE address WITH p.address
    REPLACE city WITH p.city
    REPLACE state WITH p.state
    REPLACE zipcode WITH p.zipcode
    REPLACE phone WITH p.phone
    REPLACE contact WITH p.contact
    REPLACE pur:id WITH p.pur:id
    SELECT PRIMARY
    STORE STR(5) TO number

```

```

        DELETE RECORD &NUMBER
    ENDIF
ENDDO deletes
@ 22,0 SAY CHR(7)
@ 22,4 SAY 'There Will be a Delay While the File is Packed'
SELECT PRIMARY
PACK
*PRINT THE DELETED RECORDS
    SELECT SECONDARY
    GO TOP
    @ 22,9 SAY '>>>>> Please Set Printer To Top Of Form <<<<<
    *GET READY
    WAIT
    SET FORMAT TO PRINT
    SET CONSOLE OFF
    SET HEADING TO DEPARTMENT OF SURGERY AND MEDICINE
    REPORT FORM vendet TO PRINT
    EJECT
    SET FORMAT TO SCREEN
    SET CONSOLE ON
USE ovendor
DELE ALL
PACK
RELEASE deletes, answer, reply, qoner, number
RELEASE temp, ans, numbers, i
RETURN
*****END VENDET.PRG*****

```

```

****THIS PROGRAM MODIFIES THE INFORMATION ON THE VENDOR FILE
USE INVENDOR INDEX INVENDOR
STORE T TO changes
DO WHILE changes
  STORE ' ' TO goner
  @ 10,17 GET goner
  @ 21,16 SAY '
  @ 22,0 SAY '
  READ
  IF goner = '99999'
    STORE F TO changes
    LOOP
  ENDIF
  FIND &goner
  IF # = 0
    @ 21,0 SAY CHR(7)
    @ 21,16 SAY 'No Such Vendor Code: TRY AGAIN'
    WAIT
    LOOP
  ENDIF
  *SHOW RECORD CONTENTS
  @ 11,17 SAY name
  @ 12,17 SAY address
  @ 13,17 SAY city
  @ 14,17 SAY state
  @ 15,17 SAY zipcode
  @ 16,17 SAY phone
  @ 17,17 SAY contact
  @ 18,17 SAY pur:id
  *PROMPT, CONFIRM TO CHANGE
  STORE ' ' TO answer
  @ 20,48 GET answer
  READ
  IF !(answer) = 'Y'
    STORE T TO continue
    DO WHILE continue
      STORE ' ' TO number
      @ 22,16 SAY 'Enter The Number Of Selection: '
      @ 23,16 SAY 'Use '99' TO Exit'
      @ 22,48 GET number
      READ
      DO CASE
        CASE number = '02'
          STORE ' ' TO temp
          @ 11,17 GET temp
          READ
          REPLACE name WITH temp
        *DUMMY EXTENDER
        CASE number = '03'
          STORE ' ' TO temp
          @ 12,17 GET temp
          READ
          REPLACE address WITH temp
        CASE number = '04'

```

```

STORE ' ' TO temp
@ 13,17 GET temp
READ
REPLACE city WITH temp
CASE number = '05'
STORE ' ' TO temp
@ 14,17 GET temp
READ
REPLACE state WITH temp
CASE number = '06'
STORE ' ' TO temp
@ 15,17 GET temp
READ
REPLACE zipcode WITH temp
CASE number = '07'
STORE ' ' TO temp
@ 16,17 GET temp PICTURE '999-999-9999'
READ
REPLACE phone WITH temp
CASE number = '08'
STORE ' ' TO temp
@ 17,17 GET temp
READ
REPLACE contact WITH temp
CASE number = '09'
STORE ' ' TO temp
@ 18,17 GET temp
READ
REPLACE pur:id WITH temp
CASE number = '99'
@ 22,16 SAY '
@ 23,16 SAY '
STORE F TO continue
ENDCASE
ENDDO continue
ENDIF
ENDDO changes
RELEASE changes, goner, continue, answer, number, temp
RETURN
*****END VENCHG.PRG*****

```

```

****THIS PROGRAM PROVIDES THE INFORMATION ON THE VENDOR FILE
USE invendor INDEX invendor
STORE '      ' TO goner
STORE T TO queries
DO WHILE queries
  @ 10,17 GET goner
  @ 22,16 SAY '
  @ 23,0 SAY '
  READ
  IF goner = '99999'
    STORE F TO queries
    LOOP
  ENDIF
  FIND &goner
  IF # = 0
    @ 22,0 SAY CHR(7)
    @ 22,16 SAY 'No Such Item Code: Try Again'
    WAIT
    LOOP
  ENDIF
  *SHOW RECORD CONTENTS
  @ 11,17 SAY name
  @ 12,17 SAY address
  @ 13,17 SAY city
  @ 14,17 SAY state
  @ 15,17 SAY zipcode
  @ 16,17 SAY phone
  @ 17,17 SAY contact
  @ 18,17 SAY pur:id
ENDDO queries
RELEASE queries, goner
RETURN
*****END VENQU.PRG*****

```

****THIS PROGRAM PROVIDES A TRANSACTION MENU FOR INVENTORY SYSTEM

```

SELECT PRIMARY
USE invitem INDEX invitem
SELECT SECONDARY
USE invtran
STORE T TO entering
DO WHILE entering
  *PRESENT A SCREEN OF CHOICES
  DO tranmenu
  *PROMPT, ACCEPT AN ANSWER
  STORE ' ' TO answer
  @ 8,50 GET answer
  READ
  *BRANCH TO APPROPRIATE SUPPORT ROUTINE
  DO CASE
    CASE !(answer)='SA'
      DO invtrn
      LOOP
    CASE !(answer)='AD'
      DO invtrn
      LOOP
    CASE !(answer)='PO'
      DO invtrn
      LOOP
    CASE !(answer)='RI'
      DO invtrn
      LOOP
    CASE !(answer)='PR'
      DO prtrn
      LOOP
    CASE !(answer)='RE'
      STORE F TO entering
  ENDCASE
ENDDO entering
*CLEAN UP AND QUIT
RELEASE entering, answer
ERASE
*PRESENT MAINMENU
DO invmenu
RETURN

```

*****END INVET. PRG*****

****THIS PROGRAM ACCEPTS TRANSACTIONS AND UPDATES THE ITEM FILE

STORE T TO transacts

DO WHILE transacts

*INITIALIZATION

@ 18,40 SAY '

@ 22,23 SAY '

@ 23, 0 SAY '

STORE ' ' TO goner

STORE ' ' TO t:id

STORE ' ' TO t:case

STORE 0 TO t:qty

STORE 000.00 TO total

@ 11,21 GET goner

READ

SELECT PRIMARY

IF goner = '99999'

STORE F TO transacts

LOOP

ENDIF

FIND &goner

IF # = 0

@ 22,0 say CHR(7)

@ 22,23 SAY 'No Such Item-Code: TRY AGAIN'

WAIT

LOOP

ENDIF

@ 12,21 SAY item:name

@ 13,21 SAY item:desc

@ 14,21 SAY generic

@ 18,21 SAY ret:cost

STORE ret:cost TO t:cost

CLEAR GET

*GET VARIABLES

IF !(answer) <> 'AD'

@ 15,21 get t:id

READ

ENDIF

@ 16,21 GET t:case

@ 17,21 GET t:qty

READ

IF !(answer)='SA' .OR. !(answer)='AD'

STORE t:qty * t:cost TO total

@ 18,40 SAY total USING '\$\$\$\$\$.99'

ENDIF

IF !(answer) = 'AD'

@ 19,21 GET t:remark

READ

ENDIF

STORE ' ' TO ans

@ 21,50 GET ans

READ

IF !(ans) = 'Y'

DO CASE

CASE !(answer)='SA'


```

    REPLACE q:o:h WITH q:o:h - t:qty
    REPLACE ytd:usage WITH ytd:usage + t:qty
CASE !(answer)='AD'
    REPLACE q:o:h WITH q:o:h - t:qty
CASE !(answer)='PO'
    *REPLACE reord:amt WITH t:qty
    REPLACE date:ord WITH $(tdate,7,2)+$(tdate,1,2)+$(tdate,4,2)
CASE !(answer)='RI'
    REPLACE dlastrec WITH $(tdate,7,2)+$(tdate,1,2)+$(tdate,4,2)
    REPLACE q:o:h WITH q:o:h + t:qty
    *REPLACE reord:amt WITH reord:amt - t:qty
    *IF reord:amt <= 0
        *REPLACE date:ord WITH ' '
    ENDIF
ENDCASE
SELECT SECONDARY
APPEND BLANK
*REPLACE VARIABLES IN TRAN_FILE
IF !(answer) = 'AD'
    REPLACE remarks WITH t:remark
ELSE
    REPLACE id:no WITH !(t:id)
ENDIF
REPLACE tran:no WITH t:case
REPLACE amt WITH total
REPLACE quantity WITH t:qty
REPLACE item:code WITH goner
REPLACE tran:code WITH !(answer)
REPLACE tran:date WITH $(tdate,7,2)+$(tdate,1,2)+$(tdate,4,2)
ELSE
    LOOP
ENDIF
ENDDO transacts
RELEASE ans, goner, transacts
RELEASE ALL LIKE t*
RETURN
*****END INVTNR.PRG*****

```

```

****THIS PROGRAM PROVIDES A DAILY TRANSACTION REPORT
SELECT SECONDARY
INDEX ON tran:code+item:code TO INVTRAN
USE invtran INDEX invtran
STORE ' ' TO t:code
STORE 0 TO tcount
*SET THE PRINTER
  @ 22,23 SAY 'PLEASE SET PRINTER TO TOP OF FORM'
  WAIT
  @ 22,23 SAY '***** ONE MOMENT PLEASE *****'
  @ 23,0 SAY ' '
  SET FORMAT TO PRINT
  SET CONSOLE OFF
DO WHILE .not. EOF
  IF tran:code <> t:code
    STORE tran:code TO t:code
    DO tranhead
  ENDIF
  STORE item:code TO t:item
  STORE tcount+1 TO tcount
  SELECT PRIMARY
  FIND &t:item
  IF #=0
    STORE '***TRANSACTION ERROR***' TO t:name
  ELSE
    *CAPTURE ITEM INFO FROM PRIMARY
    STORE item:desc TO t:desc
    STORE item:name TO t:name
    STORE ret:cost TO t:cost
  ENDIF
  SELECT SECONDARY
  *PRINT THAT TRANSACTION
  IF i >= 60
    DO tranhead
  ENDIF
  @ i,2 SAY t:item
  @ i,10 SAY t:name
  @ i,42 SAY t:desc
  DO CASE
    CASE t:code = 'AD'
      @ i,74 SAY tran:no
      @ i,84 SAY quantity
      @ i,93 SAY t:cost USING '$$$$.99'
      STORE t:cost * quantity TO total
      STORE total+total1 TO total1
      @ i,101 SAY total USING '$$$$$.99'
      @ i,112 SAY remarks
    CASE t:code = 'RI' .OR. t:code = 'PO'
      @ i,74 SAY id:no
      @ i,85 SAY tran:no
      @ i,99 SAY quantity
    CASE t:code = 'SA'
      @ i,75 SAY id:no
      @ i,86 SAY tran:no

```

```

    @ i,98 SAY quantity
    @ i,110 SAY t:cost USING '$$$$.99'
    STORE t:cost * quantity TO total
    STORE total+total2 TO total2
    @ i,120 SAY total USING '$$$$$.99'
  ENDCASE
  STORE i+1 TO i
  SKIP
ENDDO eof
*PRINT SUMMARY
STORE 10 TO i
@ i,35 SAY 'SUMMARY OF TRANSACTION MADE ON:'
@ i,68 SAY tdate
STORE i+3 TO i
@ i,20 SAY 'TOTAL NUMBER OF TRANSACTIONS. . . .'
@ i,56 SAY tcount USING '99999'
STORE i+2 TO i
@ i,20 SAY 'TOTAL SALES. . . . . '
@ i,56 SAY total2 USING '$$$$$.99'
STORE i+2 TO i
@ i,20 SAY 'TOTAL ADJUSTMENTS. . . . . '
@ i,56 SAY total1 USING '$$$$$.99'
*RESET THE PRINTER
  EJECT
  SET FORMAT TO SCREEN
  SET CONSOLE ON
*DELETE FILES
  USE motran INDEX motran
  APPEND FROM invtran
  DELE FILE invtran.ndx
  USE invtran
  DELE ALL
  PACK
RELEASE i
RELEASE ALL LIKE t*
RETURN
*****END PRTRN. PRG*****

```

```
****THIS PROGRAM PROVIDES A END OF PERIOD MENU FOR INVENTORY SYSTEM
STORE T TO processing
DO WHILE processing
  *PRESENT A SCREEN OF CHOICES
  DO eopmenu
  *PROMPT, ACCEPT AN ANSWER
  STORE ' ' TO answer
  @ 6,50 GET answer
  READ
  *BRANCH TO APPROPRIATE SUPPORT ROUTINE
  DO CASE
    CASE !(answer)='UH'
      DO invhist
      LOOP
    CASE !(answer)='HR'
      DO histrp
      LOOP
    CASE !(answer)='MO'
      DO invmoend
      LOOP
    CASE !(answer)='RE'
      STORE F TO processing
  ENDCASE
ENDDO processing
*CLEAN UP AND QUIT
RELEASE processing, answer
ERASE
*PRESENT MAINMENU
DO invmenu
RETURN
*****END INVEOP.PRG*****
```

```

****THIS PROGRAM UPDATES THE HISTORY FILE
*PROMPT THE MESSAGE
  @ 10,16 SAY '*** Updating the History File in Progress ***'
  @ 12,16 SAY ' Program Will Take A Few Minutes To Complete '
  @ 14,16 SAY '**** Do Not Interrupt While In Proqress ****'
  @ 16,16 SAY ' Program Will Display
*SET-UP FILES
SELECT PRIMARY
USE invhist INDEX invhist
SELECT SECONDARY
USE motran
GO TOP
DO WHILE .NOT. EOF
  IF tran:code = 'SA' .AND. upd:flag=' '
    *INIT TEMP VARIABLES
      STORE 0 TO tptd, tytd
      STORE 0.00 TO tytd:amt
      STORE item:code TO tkey
      STORE id:no TO twho
      STORE quantity TO tqty
      STORE amt TO tamt
      REPLACE upd:flag WITH 'Y'
      SELECT PRIMARY
      FIND &tkey
      IF # = 0
        *add a record to custfile
        APPEND BLANK
        REPLACE p.item:code WITH tkey
        STORE twho+str(tqty,6)+str(tqty,9)+str(tamt*100,10) TO temp
        REPLACE &twho WITH temp
      ELSE
        *update the old record
        STORE &twho TO temp
        STORE val($(temp,6,6)) TO tptd
        STORE val($(temp,12,9)) TO tytd
        STORE val($(temp,21,10))/100.00 TO tytd:amt
        *UPDATE THE USAGE
        STORE tptd+tqty TO tptd
        STORE tytd+tqty TO tytd
        STORE (tytd:amt+tamt)*100 TO tytd:amt
        STORE twho+str(tptd,6)+str(tytd,9)+str(tytd:amt,10) TO temp
        REPLACE &twho WITH temp
      ENDIF
      SELECT SECONDARY
    ENDIF
    SKIP
  ENDDO
  *CLEAN UP & RETURN
  RELEASE ALL LIKE t*
  ERASE
  RETURN
*****END INVHIST.PRG*****

```

```

****THIS PROGRAM PROVIDES A HISTORY REPORT AT THE END OF PERIOD
*GET CUSTOMER ID
  @ 8,14 SAY 'From Which Customer Do You Wish To Print Report'
  @ 9,14 SAY '<Enter Customer Id Or Press Return For All>:'
  STORE ' ' TO tid
  @ 9,61 GET tid
  STORE !(tid) TO tid
*PROMPT THE MESSAGE
  @ 11,16 SAY '**** Please Set Printer To Top Of Form ****'
  @ 13,16 SAY ' Program Will Take A Few Minutes To Complete '
  @ 15,16 SAY '**** Do Not Interrupt While In Progress ****'
  @ 17,16 SAY ' Program Will Display
  WAIT
*SET-UP FILES
USE invcust INDEX invcust
SELECT SECONDARY
USE invhist INDEX invhist
STORE ' ' TO tcust1
STORE 0 TO total1
STORE T TO printing
SET FORMAT TO PRINT
SET CONSOLE OFF
DO WHILE printing
  STORE 0 TO total1, total2
  SELECT PRIMARY
  GO TOP
  IF tid <> ' '
    FIND &tid
  ENDIF
*LOOP ON PRIMARY FILE
DO WHILE .NOT. EOF
  STORE cust:id TO twho
  STORE cust:name TO tname
  STORE 0 TO total
  DO histhead
  SELECT SECONDARY
  USE invhist INDEX invhist
  GO TOP
  DO WHILE .NOT. eof
    STORE &twho TO temp
    IF temp = ' '
      SKIP
    LOOP
  ENDIF
  STORE VAL($(temp,6,6)) TO tptd
  STORE VAL($(temp,12,9)) TO tytd
  STORE VAL($(temp,21,10))/100.00 TO tytd:amt
  STORE item:code TO tcode
  SELECT PRIMARY
  USE invitem INDEX invitem
  FIND &tcode
  IF # = 0
    STORE '***ERROR ITEM IS NOT FOUND' TO tgeneric
  ELSE

```

```

STORE generic TO tgeneric
STORE item:name TO ttrade
STORE item:desc TO tdesc
STORE ret:cost TO tcost
ENDIF
*PRINT THIS RECORD
IF i >= 60
DO histhead
ENDIF
*PRINT DETAIL LINE
@ i,2 SAY tcode
@ i,10 SAY tgeneric
@ i,52 SAY tdesc
@ i,87 SAY tptd USING '9999'
@ i,97 SAY tytd USING '99999'
*CALC TOTAL AMT
STORE tptd * tcost TO total
STORE total + total1 TO total1
STORE tytd:amt + total2 TO total2
@ i,107 SAY total USING '$$$$$.99'
@ i,120 SAY tytd:amt USING '$$$$$.99'
STORE i+1 TO i
STORE 0 TO total
SELECT SECONARY
SKIP
ENDDO eof-sec
*PRINT TOTAL FOR THIS CUST
STORE i+2 TO i
@ i,89 SAY 'GRAND TOTAL:'
@ i,106 SAY total1 USING '$$$$$.99'
@ i,119 SAY total2 USING '$$$$$$.99'
STORE 0 TO total1, total2
SELECT PRIMARY
IF tid <> ' '
GO BOTTOM
ENDIF
SKIP
ENDDO eof-pri
STORE F TO printing
ENDDO printing
RELEASE ALL LIKE t*
RELEASE printing
SET FORMAT TO SCREEN
SET CONSOLE ON
*CLEAN UP & RETURN
ERASE
RETURN
*****END HISTRP.PRG*****

```

****THIS PROGRAM PREPARES THE FILES FOR NEW PERIOD

*PROMPT THE MESSAGE

@ 10,16 SAY '***** Month End Processing Begin *****'

@ 12,16 SAY ' Program Will Take A Few Minutes To Complete '

@ 14,16 SAY '***** Do Not Interrupt While In Progress *****'

@ 16,16 SAY ' Program Will Display

USE invcust INDEX invcust

SELECT SECONDARY

USE invhist INDEX invhist

SELECT PRIMARY

GO TOP

STORE ' ' TO zeroes

DO WHILE .NOT. eof

STORE cust:id TO tcust

SELECT SECONDARY

REPLACE ALL &tcust WITH;

\$(&tcust,1,5)+zeroes+\$(&tcust,12,9)+\$(&tcust,21,10)

SELECT PRIMARY

SKIP

ENDDO eof-pri

ERASE

RETURN

*****END INVMOEND.PRG*****


```

****THIS PROGRAM PROVIDES A FACILITY TO BACK-UP THE INVENTORY FILES
*MENU COMPONENTS
DO bkupmenu
STORE T TO copying
DO WHILE copying
  STORE ' ' TO choice
  @ 7,48 GET choice
  READ
  CLEAR GET
  DO CASE
    CASE !(choice) = 'IF'
      @ 10,16 SAY '***Please Insert Item File Diskette In Drive A***'
      @ 12,21 SAY 'Press Return When Ready Or 'N' To Abort:'
      STORE ' ' TO ready
      @ 12,60 GET ready
      READ
      IF !(ready) = ' '
        @ 14,21 SAY 'Copying Item File In Progress . . .'
        USE invitem INDEX invitem
        COPY TO A:invitem.dbf
        COPY TO A:invname.ndx
        COPY TO A:invitem.ndx
      ENDIF
    CASE !(choice) = 'HF'
      @ 10,14 SAY '***Please Insert History File Diskette In Drive A**'
      @ 12,21 SAY ' Press Return When Ready Or 'N' To Abort: '
      STORE ' ' TO ready
      @ 12,64 GET ready
      READ
      IF !(ready) = ' '
        @ 14,21 SAY 'Copying History File In Progress . . .'
        USE invhist INDEX invhist
        COPY TO A:invhist.dbf
        COPY TO A:invhist.ndx
      ENDIF
    CASE !(choice) = 'VF'
      @ 10,14 SAY '***Please Insert Vendor File Diskette In Drive A **'
      @ 12,21 SAY ' Press Return When Ready Or 'N' To Abort: '
      STORE ' ' TO ready
      @ 12,64 GET ready
      READ
      IF !(ready) = ' '
        @ 14,21 SAY 'Copying Vendor File In Progress . . .'
        USE invendor INDEX invendor
        COPY TO A:invendor.dbf
        COPY TO A:vrddname.ndx
        COPY TO A:invendor.ndx
      ENDIF
    CASE !(choice) = 'CF'
      @ 10,13 SAY '***Please Insert Customer File Diskette In Drive A*'
      @ 12,21 SAY ' Press Return When Ready Or 'N' To Abort: '
      STORE ' ' TO ready
      @ 12,64 GET ready
      READ

```

```
IF !(ready) = ' '
  @ 14,21 SAY 'Copying Customer File In Progress . . .'
  USE invcust INDEX invcust
  COPY TO A:invcust.dbf
  COPY TO A:invcust.ndx
ENDIF
CASE !(choice) = 'RE'
  STORE F TO copying
ENDCASE
ERASE
DO bkupmenu
LOOP
ENDDO copying
RELEASE copying, choice, ready
DO invmenu
RETURN
*****END INVBK.PRG*****
```

```

****THIS PROGRAM PROVIDES THE INVENTORY REPORT MENU
STORE T TO reporting
DO WHILE reporting
  *PRESENT RPMENU
  ERASE
  DO rpmenu
  *ACCEPT A CHOICE OF A REPORT OR TO QUIT
  @ 9,18 SAY 'Which Report Do You Wish To Select?: '
  STORE ' ' TO report
  @ 9,56 GET report
  READ
DO CASE
  CASE !(report)='IC'
    DO invlist
    LOOP
  CASE !(report)='PI'
    DO invsheet
    LOOP
  CASE !(report)='LV'
    DO venlist
    LOOP
  CASE !(report)='RA'
    DO invreord
    LOOP
  CASE !(report)='EX'
    DO invexp
    LOOP
  CASE !(report)='RE'
    STORE F TO reporting
ENDCASE
ENDDO
*CLEAN UP AND QUIT
RELEASE reporting, report
ERASE
DO invmenu
*****END INVREPT.PRG*****

```

```

****THIS PROGRAM PROVIDES A PHYSICAL INVENTORY WORKSHEET REPORT
*GET PRINTER READY
  @ 11,16 SAY '**** Please Set Printer To Top Of Form ****'
  @ 13,16 SAY ' Program Will Take A Few Minutes To Complete '
  @ 15,16 SAY '**** Do Not Interrupt While In Progress ****'
  @ 17,16 SAY ' Program Will Display Menu When Completed. '
  WAIT
SET FORMAT TO PRINT
SET CONSOLE OFF
*PRINT HEADINGS5
  STORE 2 TO i
  @ i,50 SAY 'COLLEGE OF VETERINARY MEDICINE'
  STORE i+1 TO i
  @ i,46 SAY 'DEPARTMENT OF SURGERY AND MEDICINE'
  STORE i+1 TO i
  @ i,50 SAY 'PHYSICAL INVENTORY WORKSHEET'
  STORE i+1 TO i
  @ i,54 SAY 'DATE PRINTED: '
  @ i,68 SAY tdate
  STORE i+2 TO i
DO head5
USE invitem
INDEX ON generic TO invname
USE invitem INDEX invname
GO TOP
DO WHILE .NOT. eof
  IF i >= 59
    STORE 2 TO i
    DO head5
  ENDIF
  *PRINT DETAIL LINE
  @ i,11 SAY item:code
  @ i,20 SAY item:name
  @ i,66 SAY location
  @ i,77 SAY q:o:h
  STORE i+1 TO i
  @ i,20 SAY item:desc
  STORE i+1 TO i
  *IF generic <> '
    @ i,20 SAY generic
    STORE i+1 TO i
  ENDIF
  SKIP
ENDDO eof
*CLEAR UP AND QUIT
  SET CONSOLE ON
  SET FORMAT TO SCREEN
RELEASE i
RETURN
*****END INVSHEET.PRG*****

```

****THIS PROGRAM PROVIDES A LIST OF INVENTORY ITEM

*GET PRINTER READY

@ 11,16 SAY '**** Please Set Printer To Top Of Form ****'

@ 13,16 SAY ' Program Will Take A Few Minutes To Complete '

@ 15,16 SAY '**** Do Not Interrupt While In Progress ****'

@ 17,16 SAY ' Program Will Display Menu When Completed. '

WAIT

SET FORMAT TO PRINT

SET CONSOLE OFF

*PRINT HEADING1

STORE 2 TO i

@ i,50 SAY 'COLLEGE OF VETERINARY MEDICINE'

STORE i+1 TO i

@ i,46 SAY 'DEPARTMENT OF SURGERY AND MEDICINE'

STORE i+1 TO i

@ i,48 SAY 'LIST OF ITEMS IN THE INVENTORY'

STORE i+1 TO i

@ i,54 SAY 'DATE PRINTED: '

@ i,68 SAY tdate

STORE i+2 TO i

DO head1

USE invitem INDEX invname

GO TOP

DO WHILE .NOT. eof

IF i >= 59

STORE 2 TO i

DO head1

ENDIF

*PRINT DETAIL LINE

@ i,11 SAY item:code

@ i,20 SAY item:name

@ i,66 SAY location

@ i,77 SAY q:o:h

@ i,93 SAY ret:cost USING '\$\$.99'

STORE i+1 TO i

@ i,20 SAY item:desc

STORE i+1 TO i

*IF generic <> '

@ i,20 SAY generic

STORE i+1 TO i

ENDIF

SKIP

ENDDO eof

*CLEAN UP AND QUIT

SET CONSOLE ON

SET FORMAT TO SCREEN

RELEASE i

RETURN

*****END INVLIST.PRG*****

```

****THIS PROGRAM PROVIDES A LIST OF VENDOR FROM VENDOR FILE
*GET PRINTER READY
  @ 11,16 SAY '**** Please Set Printer To Top Of Form ****'
  @ 13,16 SAY ' Program Will Take A Few Minutes To Complete '
  @ 15,16 SAY '**** Do Not Interrupt While In Progress ****'
  @ 17,16 SAY ' Program Will Display Menu When Completed. '
  WAIT
SET FORMAT TO PRINT
SET CONSOLE OFF
*PRINT HEADING2
  STORE 2 TO i
  @ i,50 SAY 'COLLEGE OF VETERINARY MEDICINE'
  STORE i+1 TO i
  @ i,46 SAY 'DEPARTMENT OF SURGERY AND MEDICINE'
  STORE i+1 TO i
  @ i,43 SAY 'LIST OF VENDORS IN THE INVENTORY SYSTEM'
  STORE i+1 TO i
  @ i,54 SAY 'DATE PRINTED: '
  @ i,68 SAY tdate
  STORE i+2 TO i
DO head2
USE invendor
INDEX ON name TO vdrname
USE invendor INDEX vdrname
GO TOP
DO WHILE .NOT. eof
  IF i >= 59
    STORE 2 TO i
    DO head2
  ENDIF
  *PRINT DETAIL LINE
  @ i,10 SAY vendor:id
  @ i,20 SAY name
  @ i,56 SAY phone USING '999/999-9999'
  @ i,70 SAY pur:id
  @ i,80 SAY contact
  STORE i+1 TO i
  @ i,20 SAY address
  STORE i+1 TO i
  @ i,20 SAY city
  @ i,41 SAY state
  @ i,44 SAY zipcode
  STORE i+2 TO i
  SKIP
ENDDO eof
*CLEAR UP AND QUIT
  SET CONSOLE ON
  SET FORMAT TO SCREEN
RELEASE i
RETURN
*****END VENLIST.PRG*****

```

```

****THIS PROGRAM PROVIDES A LIST OF EXPIRED ITEM
*GET PRINTER READY
  @ 11,16 SAY '**** Please Set Printer To Top Of Form ****'
  @ 13,16 SAY ' Program Will Take A Few Minutes To Complete '
  @ 15,16 SAY '**** Do Not Interrupt While In Progress ****'
  @ 17,16 SAY ' Program Will Display Menu When Completed. '
  WAIT
SET FORMAT TO PRINT
SET CONSOLE OFF
*PRINT HEADING3
  STORE 2 TO i
  @ i,50 SAY 'COLLEGE OF VETERINARY MEDICINE'
  STORE i+1 TO i
  @ i,46 SAY 'DEPARTMENT OF SURGERY AND MEDICINE'
  STORE i+1 TO i
  @ i,43 SAY 'LIST OF EXPIRED ITEM IN THE INVENTORY SYSTEM'
  STORE i+1 TO i
  @ i,54 SAY 'DATE PRINTED: '
  @ i,68 SAY tdate
  STORE i+2 TO i
DO head3
STORE str(tdate,7,2)+str(tdate,1,2) TO cdate
USE invitem INDEX invitem
GO TOP
DO WHILE .NOT. eof
  IF dexpire <> '
    IF i >= 59
      STORE 2 TO i
      DO head3
    ENDIF
    IF dexpire <= cdate
      @ i,11 SAY item:code
      @ i,20 SAY item:name
      @ i,66 SAY location
      @ i,77 SAY q:o:h
      STORE str(dexpire,3,2)+'/'+str(dexpire,1,2) TO cexpire
      @ i,93 SAY cexpire
      STORE i+1 TO i
      @ i,20 SAY item:desc
      STORE i+1 TO i
      IF generic <> '
        @ i,20 SAY generic
        STORE i+1 TO i
      ENDIF
    ENDIF
  ENDIF
  SKIP
ENDDO eof
*CLEAR UP AND QUIT
SET CONSOLE ON
SET FORMAT TO SCREEN
RELEASE i
RETURN
*****END INVEXP.PRG*****

```

****THIS PROGRAM PROVIDES A INVENTORY REORDER REPORT

*GET PRINTER READY

@ 11,16 SAY '**** Please Set Printer To Top Of Form ****'

@ 13,16 SAY ' Program Will Take A Few Minutes To Complete '

@ 15,16 SAY '**** Do Not Interrupt While In Progress ****'

@ 17,16 SAY ' Program Will Display Menu When Completed. '

WAIT

SET FORMAT TO PRINT

SET CONSOLE OFF

*PRINT HEADING4

STORE 2 TO i

@ i,50 SAY 'COLLEGE OF VETERINARY MEDICINE'

STORE i+1 TO i

@ i,46 SAY 'DEPARTMENT OF SURGERY AND MEDICINE'

STORE i+1 TO i

@ i,44 SAY 'REORDER ADVICE REPORT FROM ITEM DATA BASE'

STORE i+1 TO i

@ i,54 SAY 'DATE PRINTED: '

@ i,68 SAY tdate

STORE i+2 TO i

DO head4

STORE F TO flag

USE invitem INDEX invitem

GO TOP

DO WHILE .NOT. eof

IF i >= 59

STORE 2 TO i

DO head4

ENDIF

IF q:o:h <= reord:pt

@ i,11 SAY item:code

@ i,20 SAY item:name

@ i,65 SAY q:o:h

@ i,80 SAY reord:pt

@ i,94 SAY reord:amt

@ i,105 SAY vendor1

STORE i+1 TO i

@ i,20 SAY item:desc

IF vendor2 <> '

@ i,105 SAY vendor2

ENDIF

STORE i+1 TO i

IF generic <> '

@ i,20 SAY generic

STORE T TO flag

ENDIF

IF vendor3 <> '

@ i,105 SAY vendor3

STORE T TO flag

ENDIF

IF flag = T

STORE i+1 TO i

STORE F TO flag

ENDIF


```
ENDIF
SKIP
ENDDO eof
*CLEAN UP AND QUIT
SET CONSOLE ON
SET FORMAT TO SCREEN
RELEASE i, cexpire, cdate, flag
RETURN
*****END INVREORD.PRG*****
```

****THIS ROUTINE PROVIDES A HEADING OF INVENTORY ITEM LIST

```
@ i,10 SAY 'ITEM-NO'
@ i,30 SAY 'DESCRIPTION'
@ i,65 SAY 'LOCATION          ON-HAND          UNIT-COST'
STORE i+2 TO i
RETURN
```

*****END HEAD1.PRG*****

****THIS ROUTINE PROVIDES A HEADING OF VENDOR LIST

```
@ i, 8 SAY 'VENDOR-NO'
@ i,27 SAY 'NAME & ADDRESS'
@ i,59 SAY 'PHONE          PURCHASE-ID          CONTACT'
STORE i+2 TO i
RETURN
```

*****END HEAD2.PRG*****

****THIS ROUTINE PROVIDES A HEADING OF EXPIRED ITEM LIST

```
@ i,10 SAY 'ITEM-NO'
@ i,30 SAY 'DESCRIPTION'
@ i,65 SAY 'LOCATION          ON-HAND          EXPIRED-DATE'
STORE i+2 TO i
RETURN
```

*****END HEAD3.PRG*****

****THIS ROUTINE PROVIDES A HEADING FOR REORDER REPORT

```
@ i,10 SAY 'ITEM-NO'
@ i,30 SAY 'DESCRIPTION'
@ i,65 SAY 'ONHAND          REORDER-LVL          REORDER-QTY          VENDOR-ID'
STORE i+2 TO i
RETURN
```

*****END HEAD4.PRG*****

****THIS ROUTINE PROVIDES A HEADING FOR PHYSICAL INVENTORY WORKSHEET

```
@ i,10 SAY 'ITEM-NO'
@ i,30 SAY 'DESCRIPTION'
@ i,65 SAY 'LOCATION          ON-HAND          ACTUAL-ON-HAND'
STORE i+2 TO i
RETURN
```

*****END HEAD5.PRG*****

AN INTERACTIVE INVENTORY SYSTEM
BASED ON DBASE II

by

SUTEERA S. GRAHAM

B. S., Emporia State University, 1978

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1984

ABSTRACT

The report presents the process of developing an inventory data base system. The process includes the feasibility study, the system design, the implementation and the documentation.

First, a detailed study of the function was carried out to determine the feasibility of an automated inventory. The feasibility study provided an evaluation of alternative approaches to the development of a system and allowed decisions to be made. The stages of development convert a manual file system to a relational data base. The relations were then used as a basis to define the record forms file system and analyzed for meaning and usage. The automated Warnier diagram production program is used as a system design tool. The Warnier diagram is a method for representing an information hierarchy and depicts information as a tree-like data structure. The system design was then expanded into source code statement using dBASE II as a target language to produce the system implementation. The implementation restrictions reflect the intended user group and the properties of microcomputers, including memory size and speed of access for storage devices. A user's guide is provided to document use of the system from data entry and file maintenance to report generation.