

PERSONNEL RECORD SYSTEM
A CASE STUDY OF
USMANU DANFODIYO UNIVERSITY, SOKOTO

BY

SULAIMAN ISAH

PGD/MCS/114

**SUBMITTED TO THE DEPARTMENT OF MATHEMATICS,
COMPUTER AND STATISTICS, SCHOOL OF SCIENCE AND
SCIENCE EDUCATION IN PARTIAL FULFILMENT
FOR THE AWARD OF POST GRADUATE DIPLOMA IN
COMPUTER SCIENCE**

**AT THE
FEDERAL UNIVERSITY OF TECHNOLOGY, MINA.**

JANUARY, 1996.

TABLE OF CONTENTS

CHAPTER ONE: ANALYSIS OF THE PROBLEM SYSTEM	
TITLE PAGE	I
APPROVAL PAGE	iv
DECLARATION	v
DEDICATION	vi
ACKNOWLEDGMENTS	vii
CHAPTER ONE: INTRODUCTION	
1.1 Information and Society	1
1.2 Importance of Information	2
1.3 A brief history of computers	3
1.4 Computer Generations	5
CHAPTER TWO: LITERATURE REVIEW	
2.1 Automation of manual system	9
2.2 Advantages of evolution of data processing	10
2.3 Computer applications today	12
2.4 Systems Analysis and design - Overview	17
2.5 Methods and Procedures	18
2.6 Information Systems Development	18

APPROVAL PAGE

This research project has been read and approved by the undersigned persons and have satisfied the requirements of the Department of Mathematics, Computer and Statistics, Federal University of Technology, Mina in partial fulfilment of the conditions for the award of Post Graduate Diploma in Computer Science.

Project Supervisor

Signature

Date

Head of Department

Signature

Date

External Examiner

Signature

Date

DECLARATION

I hereby declare that this project is written by me. All materials and information used in compiling this work were obtained from published books and handout and all have been acknowledged and referenced accordingly.

Name of student

Signature

Date

The above declaration is confirmed by me.

Name of Supervisor

Signature

Date

DEDICATION

I dedicate this project to my first child Abdur-Rahman, who was born during this programme, to my wife Zaliha Sulaiman who patiently waited at their home during this programme, to my loving mother Malama Hauwa Kunduda Isah and to my late father Alh. Kunduda Isah Augie, may his gentle soul rest in perfect peace amen.

ACKNOWLEDGMENTS

All praise goes to Almighty Allah the sustainer of all for His mercies and strength granted me on the course of this work and always.

I wish to express my special thanks and appreciation to my supervisor Dr. Sunday Reju for his painstaking efforts, advice and constructive criticism which made this work successful.

I find it necessarily necessary to express my profound gratitude to my sponsors Usmanu Danfodiyo University, Sokoto for given me this opportunity to further my studies.

I must express my profound gratitude to my able course coordinator Mr. Badmus for his numerous advise and contributions too numerous to mention during this work.

Special thanks goes to my wife Zaliha for her support and encouragement. May Allah continue to bless our lives.

I must ever remain grateful to the following people who contributed in one way or the other Mal. Shehu Usman Gulumbe, Mal. Ibrahim Babawotu, ASP Nura S. Tsamiya, ASP Ibrahim Zuru and many other good friends too numerous to mention here.

Sulaiman Isah
January, 1996.

CHAPTER ONE

INTRODUCTION

The importance of human efforts in any organisation can not be over emphasised. In view of the role human beings play towards the success of any organisation, it is important that comprehensive and easy accessible records are maintained for them.

There is no doubt about its flow the most efficient and reliable storage of data is that which is computer based. In Usmanu Danfodiyo University, Sokoto, a manual system of maintaining staff record is being operated. However, the over growing size of staff strength call for computer application in staff record keeping to aid management information system.

1.1 INFORMATION AND SOCIETY:

Man has, over the years, used machine and animals to lighten the burden of manual labour. Today, we use computers to lighten our burden of storing, processing, and retrieving data for decision making. A computer is a machine capable of complex systematical and logical operations and processing of voluminous data and information with enormous speed and absolute accuracy. Computers have revolutionalised the field of computing as a result of technological advancement.

When the computer was first invented, it was seen to be an efficient tool for performing scientific computation. Later it was discovered that a computer could perform a wide variety of tasks, processing and retrieval of data and information. Earlier, the use of data meant numbers, but soon computers came to be used for processing textual and graphic data as well. With the linkage between computers and communication, such data could also be transmitted across space. Later they began to have impact on the functioning of commercial organisation and the government by processing large amount of data and

converting it into managerially useful information. Today, their impact is all pervasive in education, entertainment, trade and banking, perhaps in every sphere of human activity.

1.2 IMPORTANCE OF INFORMATION:

There are three principal reasons why information systems are subject to great interest today.

- (1) First, Organisations have grown to unprecedented complex levels. Information plays a vital role in holding together and coordination organisation. It is the mortar that holds together the edifices of modern multi division, multi location, multi product organisation on the role of information system. It is very similar to that of the nervous system in animals.
- (2) The second reason is the advent of the computer. Computers are able to both access and record information and perform calculation at speed which are truly unbelievable. Computers could access information at a rate higher than 20,000 characters per second from magnetic tape, which could correspond to readings and writing about 400 pages of an oversized book in one minute.

When it comes to calculation computer can do about 50,000 multiplication in one second. It is a unique feature of computer technology that between its arrival on the management scene in the mid-fifties and its development, the cost of information processing has decreased by at least two orders of magnitude.

- (3) The third reason is the progress made in communication networks. Such networks allow the sharing of EXPENSIVE CPU (Central Processing Unit) resources, discs, printers and other hardware. It has now become economic to transmit large amount of information across long distances almost instantaneously.

1.3 A BRIEF HISTORY OF COMPUTERS

1.3.1 Mechanical Calculators

The history of computer is longer than one might think, with Blaise Pascal (1623-1662) developing a mechanical calculator known as the Pascaline, which based upon gears. This reflected the technology of the moment. The Pascaline was capable of addition and subtraction upon whole numbers.

Gottfried Wilhelm Von Leibnitz (1646-1716) improved upon the Pascaline by developing a machine which could also perform multiplication. However, it was too complex for the available technology and proved unreliable. technological limitations have constrained the development of computers throughout their history and it is only within the past thirty years that the technology has progressed sufficiently quickly to avoid limiting their progress. Charles Babbage (1792-1872) was an eminent Mathematician who become increasingly frustrated with the inaccuracies inherent in his work. These inaccuracies were particularly prevalent in the log tables he used for multiplication and division and he became convinced that a machine could derive the tables more accurately. The system which he developed was known as the difference machine.

There are many technical difficulties associated with the difference machine, but Babbage moved on to design an even more sophisticated system known as the analytical

engine. This had a number of important features, including the following:

1. It was a general-purpose machine, that it was able to perform any calculation.
2. Information was entered using punched cards, which had been developed earlier for weaving machines by Joseph Jacquard (1752-1834). A variation of these punched cards is still used with some mainframe computers.
3. The machine had an output device.
4. The data entered on the punched cards was a simple program of a similar type to those used in present day computers.

All of these calculating machines up till this time had been either single machines or limited production models.

1.3.2. Electromechanical Computers

Until this point the machines had mainly been hand driven. With the widespread availability of electrical energy a new approach was developed based around an electrical switch known as a relay. A relay consists of a coil wire and two contacts. When current is passed they are open. This means that relays can respond very quickly to a change in the current, consequently resulting in a faster machine.

The first electromechanical computer was the Harvard Mark 1, developed by Professor Howard H. Aiken (1900-1973) of Harvard University under the sponsorship of International Business Machines (IBM). This machine, was completed in 1944, developed upon Babbage's idea of the computer being programmable. It operated relatively quickly, but it still took 3 seconds to multiply two numbers. It was also very large, measuring 51 feet and containing 500 miles of wire.

1.4 COMPUTER GENERATIONS

Generally, advances in computer technology can be classified into categories called computer generations. What distinguishes each generation is the main electronic logic element in use at that time. The term logic element refers to the electronic components used to facilitate the circuit functions within the computer.

1.4.1 First generation computers: Valves (1950-1958)

The Harvard Mark 1 was overtaken in 1946 by the ENIAC (Electronic Numerical Integrator and Computer) machine which was based around valve technology. Valve technology developed during the second world war as a by-product of the research into radars. The valve, like the relay, is a form of switch. However it is an electronic switch, with no moving parts and consequently operates more quickly. ENIAC contained 18,000 valves and weighed over 30 tons. It was able to perform multiplication in 3 milliseconds (3/1000 second).

The successor to ENIAC WAS EDVAC (Electronic Discrete Variable Automatic Computer) which was developed by John Von Neuman from Princeton's Institute for Advanced Studies and completed in 1950. Von Neuman was the first person to have the idea of a stored program: strong the program inside the computer itself. EDVAC implemented this idea and also took advantage of the two states inherent in the operation of valves. It was therefore based around the binary number system.

During the 1950s, valve computers developed and improved. However, valves were inherently unreliable because of the temperature at which they operated and it was a time consuming process simply ensuring that the computer continued to function. The programs were typically entered on punched cards and stored in magnetic drum memory, which is a

cylinder with a magnetised outer surface. The programs were all written in machine code. It was only towards the end of the 1950s that slightly simpler form of programming language, known as assembly language, was developed.

1.4.2 Second generation computers: transistors (1959-64)

In 1948, in the Bell Telephone Laboratories, John Bardeen, Walter Brattain and William Shockley invented the transistor. The transistor, like the relay and valve was a type of a switch, but it was superior to both as it was small, ran at low temperatures and was more reliable. Transistors were constructed from either germanium or silicon. The second generation of computers adopted this technology.

Two new types of memory were also developed, known as a magnetic core memory and magnetic disk storage. Magnetic core was used for main memory and magnetic disk for secondary storage. During this generation a new type of programming language was developed, called the high-level language. These include BASIC, FORTRAN and COBOL. Each of these languages was developed for a different purpose: FORTRAN for solving mathematical problems, COBOL for writing business Programs and BASIC for beginners to the computer. There is still no Universal Programming language which is suitable for all problems and since the original development of high-level programming languages. The increase in reliability of this generation of computers and the development of easier-to-use programming languages meant that the use of computers for business and research increased considerably.

1.4.3 Third generation computers: Integrated circuits (1965-1971)

Central to the development of the third generation of computers was the discovery

that a number of transistors and connections between them could be imprinted on a single slice of silicon. This was referred to as an integrated circuit. Integrated circuits were smaller, cheaper, more reliable and faster than circuits made of individual transistors. As a result of this development it was possible to build more powerful computers which were also smaller and cheaper. In addition to the CPU being made of integrated circuits, so was the main memory. By 1969 it was possible to place 100 transistors on a single piece of silicon.

The software available for this generation of computers was also becoming more sophisticated. The development of Operating Systems and time sharing were characteristic of this generation. Many of the existing mainframe computers originate from this period.

1.4.4 Fourth generation computers: Large/very large Scale Integration (1971 - present)

Since the development of integrated circuits, the number of transistors which can be placed on a slice of silicon has increased rapidly. In the early 1970s, LSI (Large Scale Integration) chips were developed which contain several thousand transistors and by the mid-1970s VLSI (Very Large Scale Integration) chips were capable of containing the whole CPU on a single chip, called microprocessor. The microprocessor forms the basis all present day microcomputers (hence the name). As a consequence throughout the 1970s, computers became smaller and cheaper.

As with the development of the computer in the 1960s and 1970s, there has been a very rapid improvement in the microprocessor technology during the 1980s. As a consequence, their operation has become faster and they are more flexible.

Central to the development has been the establishing of an industry standard IBM in the

IBM PC (Personal computer). The IBM PC was a cheap, but reasonably powerful and versatile microcomputer which was widely adopted by industry. It used an operating system called MS-DOS. Such computers can now be found both in industry and education. As the microprocessor technology improves, these computers become faster, with larger memories, but they are designed to be "upward compatible", with each new version capable of running the software of the previous.

1.4.5 The future

In the short term the popularity of PC computers and Apple Macintosh's seems likely to continue. However, the nature of these machines is altering with the technology and they are becoming similar in power to workstations.

In the long term, the trend towards easier to use computers, which started with the development of high-level computer languages, is likely to continue. For example, the windowing environments. It is also likely that more sophisticated programs will be developed to enable more flexible input and output, allowing for the recognition and generation of speech and the recognition of visual images. These areas are sub-areas of artificial intelligence, the part of computer science which is concerned with the emulation of human intelligence in computer systems.

Finally, it is difficult to anticipate hardware developments, as in the recent history they have been so rapid and dramatic. However, it seems likely that the mass storage medium, CD-ROMS, which are the same size as an audio compact disk and can contain a whole encyclopaedia of data, will before increasingly important.

CHAPTER TWO

EVOLUTION OF THE USE OF COMPUTERS IN AN ORGANISATION

INTRODUCTION

Most organisations carry out a large and wide variety of business transaction. Accurate recording and processing of these transaction is known as DATA PROCESSING. Today, some degree of automation exists in organisation in recording and processing of daily transaction. This may be in the form of calculating aids, mechanical/electronic data capture systems, and computers which contain integrated devices for capturing, storing, and retrieving of data. However, thee decades ago and even in some organisation, transaction processing systems were largely manual.

2.1 AUTOMATION OF MANUAL SYSTEMS

As the complexity and size of organisation grew, so did the number of transactions processed. Manual systems were then devised to record the transaction on documents. These documents were subsequently processed by clerical staff to produce ledgers reflecting the impacts of transaction on items so interest to managers. Example include accounts ledgers to provide daily store ledgers and cards to provide inventory balances of each item.

The clerical systems designed were ideal to take care of average loads of posting transaction. As long as the number of financial transaction/store transaction in a month were in hundreds, it was possible to close accounts and know the financial performance of a business at the end of the month or at the end of the year. However, business activities being often uneven, clerks were saddled with backlog of transactions to be poshed. This resulted in vouchers getting misplaced and made the posting systems error prone.

Enormous time was also required to clear the backlog and trace and correct mistakes made earlier. When organisations grew, manual systems began to break down, this hinders management in the primary task of planning and control. In western countries clerical costs also began to rise and became significant. Thus, there was the need to keep the clerical costs low and to increase efficiency of processing transaction. A genuine need was felt (and justified) to mechanise data processing.

In the sixties and early seventies computers were introduced for processing transactions that arose in various business transactions. These computers offered input/output peripherals which were rugged and could handle large volume of data. Their processing capability was limited compared to modern computers. Such computers were seen as an ideal solution for the data processing tasks which required handling large input/output but limited processing.

2.2 ADVANTAGES OF EVOLUTION OF DATA PROCESSING

Business and government establishments found that they could computerise the transaction processing tasks quickly by using standard software. The major advantages of such software were:

- (a) Minimised delays in processing transaction.
- (b) Improved accuracy.
- (c) Ability to cope with future growth.

(a) **Minimised delays:** In computerised processing, data capture and preparation were labour intensive while data processing was computerised. The uneven transaction load of individual departments when pooled together evened the load of data entry. This led to

minimising the delays on account of data preparation. Purchased cards provided a convenient form of data input. Data/information and programs purchased in cards were transmitted through a high speed card reader into a computer for processing. Thus, organisation found from their invoices could be sent a few days earlier to the customers, pay cheques could be delivered at the end of the month, and stores ledger produced within two or three days of the month end.

High speed card readers and time printers were found to be more rugged and reliable compared to equipments like typewriters used in manual processes. The advent of pre-printed stationary and multipart stationary made the task of generating various types of reports/invoice even easier.

(b) **Improved accuracy:** In manual systems, accuracy is low because of computational errors and inability to process some transactions or errors committed due to fatigue in repetitive clerical tasks. Since standard software was developed for data processing systems, it was possible to make its logic error free. It was also possible to build in various types of error checking procedure to ensure that data entering the system for processing was error free.

(c) **Low Clerical Costs:** In the developed countries where clerical manpower is costly and computer hardware/software relatively cheap, organisation have found that they can cut down clerical costs by computerised transaction processing.

(d) **Ability to cope with future growth:** Growing organisation have also found in recent years that computers enable them to cope with future growth easily without having to expand the clerical workforce. They also observed that the cost of data preparation be

in ten volume of transactions, but the processing cost of computers does not go up in the same proportion.

2.3 COMPUTER APPLICATIONS TODAY:

The use of computers is more widespread now than at any tie in the past. They pervade almost every aspect of our lives: they control our washing machine nd our microwave oven; they help to teach our children and to design our cars; our personal details are stored by computers for use by doctor, tax inspector and electricity board.

At this print, we will look at some of the main areas of computer applications and discuss briefly a few examples in each area.

2.3.1 Applications in Education:

Now that computers and their associated peripherals are relatively cheap, they have been adopted as a learning tool throughout the education system. Nowadays computers are used throughout the curriculum, whereas in the past they were confined to computer studies departments. Thus schools no longer merely teach about computers - they teach using computers as a tool.

Various different approaches have been used including: Computer Managed Learning (CML), Computer Assisted Learning (CAL), and Computer Based Training (CBT).

More recently, the emphasis has tended to be put on the development of simulations, adventure games and 'content-free' packages.

2.3.2 Scientific Modelling:

Scientists often use computers to 'model' events in their particular field of interest. For example, meteorologists make use of computers to predict the weather. Data about current atmospheric conditions is fed into a powerful computer, which performs calculations according to rules given by the scientist. These predictions are compared with the actual data for that time, when it becomes available.

2.3.3. Applications in business - data processing

Any large business is likely to have to store large amount of data; for example, the names of customers, details of employees. A computer provides a good way of storing such information.

Smaller business may use a microcomputer to store their information. Packages are now available to store and retrieve information, to keep accounts, and to produce documents and drawings.

2.3.4. Applications in manufacturing

The first use of computers in manufacturing was as an aid in the designing of new products. Initially Computer-Aided Design (CAD), was simply used as a tool to help draughtsmen produce better drawings more quickly. It was soon developed to do much more than this.

Now CAD software can display an object from any required angle; display cross-sections of any part; display the object in line-drawing or solid modelling; select appropriate standard components. The image on the screen may be stored permanently in a file, together with any additional information required. Hard copy plans can be produced using

a special output device called a plotter. A list of parts can be produced. Changes to design can be carried out easily, without the need for costly redrawing.

2.3.5 Online information retrieval systems

When using an online information retrieval system, the user gains access to a database by means of a computer terminal lined to the computer in which the database is stored. The terminal is linked to the central computer via a telecommunications network. A large variety of databases are now available, supplying a wide range of information. Such database can, therefore be used as an alternative to searching for information in a library.

2.3.6 Real-time applications

A real time system is one where the computer is required to keep pace with some external process. This often means that the computer must interact with other pieces of equipment. Examples of real-time applications include the computer-control of machinery, from our home video recorders and washing machines to the robots used in manufacturing industry. Sensors within the machine pass information about what is happening to a computer, which reacts by giving instructions as to what to do next. For example; the microcomputer controlling a washing machine might give an instruction to open the water inlet. It will then wait for a sensor to respond with the information that the water has reached a prescribed level. The computer will then issue an instruction to close the valve.

In a hospital, a patient's condition may be monitored by a computer. The computer is programmed to alert the medical staff if the patient requires attention.

On larger scale, real-time computer systems can be used to control traffic flow in a town.

2.3.7. Applications in the retail trade

An increasing number of supermarkets now have checkouts which incorporate a 'barcode reader'. As each item is passed over it, the barcode reader reads the code and passes the number to a microcomputer. The computer has stored details of the price and description of every item for sale. It instructs the printer to print out the price and description of the item on the till receipt.

Automatic checkouts of this kind enable goods to be processed much quicker than using a manual system. There is also a saving for the retailer since it no longer necessary to employ staff to stamp each item with its price.

Each checkout is connected to a larger computer which can be used to keep records of the sales of each item throughout the store.

2.3.8 Expert Systems

An expert system consists of a "knowledge base" of rules of the form: if A is true the B is true, together with software which is capable of accepting data and applying the given rules to form valid conclusions. This is just the sort of reasoning which a doctor uses when making his/her diagnosis.

A typical example of the use of an expert system is that of a doctor, who uses a computer to help him in his diagnosis. Expert systems are at present, mainly confined to the fields of medicine, science and engineering.

2.3.9 Computer-Aided Software Engineering

Computer-aided software engineering (CASE), was first developed in the late 1970s to help programmers to design software. The early CASE tools did little more than enable diagrams of the program structure to be produced. Nowadays, these tools have become more sophisticated, and they can be incorporated into an integrated environment providing documentation facilities and dictionaries of flow charts. Once the program has been designed, a CASE tool may be available which can write a proportion of the program code itself. Another may be used to check existing code.

A CASE package may also be useful for producing a prototype software package to illustrate what the package will look like when it is completed. Standard menu screens, for example, would give the customer a good idea of the choices which will be available to him without the necessity for writing code specifically for his requirements. This gives the

customer the chance to try out the program before a great deal of work has been put into it, and to suggest modifications at an early stage.

2.3.10 The paperless office:

There are a large number of uses for computer in office. Offices rarely found a use for the larger computers of the past, but today, with the advent of the "personal computer", computer are being used for a number of tasks.

Today, typewriters have often been replaced by wordprocessors, which enable documents to be prepared quickly and to a high standard.

Computer systems in different offices may be able to communicate by means of an electronic mail system. A message is typed into a computer in one office. The message can then be transmitted electronically to a different office, where it may be stored in a computer file, displayed on a screen, or printed out.

2.4 SYSTEMS ANALYSIS AND DESIGN-OVERVIEW

Systems analysis and design describes the process used by systems analysts who design information systems and data processing systems based on the information requirements vital to the development of computer based information systems. Since ineffective and inefficient use of computers in business is frequently attributed to a failure to understand and appl the systems concept to the information requirements of the organisation.

2.5 METHODS AND PROCEDURES

The systems analysis and design function in many organisation originated in "methods and procedures" groups and may still exist in "systems and procedures" departments. Methods and procedures departments applied the techniques of "scientific management" to the data processing and communication function of an organisation. Thus, systems analysis is related to scientific management and industrial engineering.

The increase in paper work and shortage of clerical employees during World War II encouraged the development of specialists in "office procedures" or "methods and procedures" in government agencies and large organisation. These specialists engage in such activities as forms design, forms control, work specification, time and motion studies, work sampling, office layout design procedure writing etc.

2.6 INFORMATION SYSTEMS DEVELOPMENT

Computer based information systems do not just happen. They must be conceived designed and implemented. Developing an information system requires a process of systems development which includes the following steps:

- (i) Investigation
- (ii) Analysis
- (iii) Design
- (iv) Programming
- (vi) Maintenance

(a) **Investigation**: The systems analysis and design process begins with the systems investigation phase which results on the selection and definition of the particular system that will be developed. Systems investigation in the stage which answers the question: "should we develop new or improve system?" or improved systems can be costly once, the system investigation stage requires a preliminary study called a feasibility study". The system investigation stage is also known as "needs research". System investigation should include the following steps; surveying, defining and selecting a system to be developed. Making a preliminary determination of the information needs of prospective users, and the objectives, constraints, basic resources requirements, cost benefits, and feasibility of the proposed system.

(b) **Analysis**: System analysis involves analysis in detail the information needs of prospective users and developing the system requirements of a proposed system. The system analysis is a "top-down" and structured series of steps summarised below, analysing in detail the information needs of prospective users and developing the system requirements of the proposed system, requiring a detailed analysis of:

- (i) the organisation that will use the system
- (ii) the system presenting used, if any; and
- (iii) the logical input/output, storage and processing requirements of the proposed system.

(c) **Design**: System design involves the development of a logical and physical design for an information system that meets the system requirements developed by the systems analysis process. System design involves the detailed design of input document, output

reports, data base and processing procedures personnel, data media, equipment and programming specification are also developed for the proposed system. The system design stage can be segmented into three major steps outlined below:

- (i) **Logical System Design:** This develops and finalizes a "logical" system design by a thorough analysis of alternative logical data flows and data processing activities.
- (ii) **Physical System Design:** Design in detail the "physical" step which includes specific input, processing, output storage and control media, methods, and procedures.
- (d) **System Specification:** Develop the "system" specifications report which documents the entire systems design and specifies the hardware, software, database, procedures, and personnel to be used by the system.
- (e) **Programming:** The programming activity involves the development of computer programs that meet the programming specification of the design stage. Programming stage requires continual interaction between the systems analysts, programmers and the computer users who may be part of a "system development project team". This is true especially during the programme analysis and programme design stages, as well as during programme verification, implementation and maintenance.
- (f) **Systems Implementation:** The activities of systems implementation involve the testing documentation acquiring, installing and operation of a newly designed system and the training of personnel to operate and use the system.
- (g) **Systems Maintenance:** Systems Maintenance involves the monitoring, evaluating, and modifying of an established system in order to make desirable or necessary improvements.

For example, errors in the operation or use of the system must be corrected by the systems maintenance activity. Installation of a new system usually result in the phenomenon called the "learning curve." Personnel who operate and use the system will make mistakes simply because they are not familiar with the new system.

The system maintenance activity requires a periodic reviewing or auditing for the system to ensure that it is operating properly and meeting its objectives. Systems maintenance also incudes making modification to a system due to changes within the organisation or in the business environment.

CHAPTER THREE

ANALYSIS OF THE PRESENT SYSTEM

3.1 A BRIEF HISTORY OF USMANU DANFODIYO UNIVERSITY SOKOTO

Usmanu Danfodiyio University, Sokoto (formerly University of Sokoto until the change of name in 1988) was one of the four universities established by the Federal Government of Nigeria in 1975, at which the time three University Colleges (now full fledged Universities) were also established. The seven Universities together are now popularly called Nigeria's Second Generation Universities or the Seven Sisters.

The University started to function on a temporary site, in what is now referred to as the city campus along Sultan Abubakar Road of Sokoto Municipality. The city campus now houses the University's College of Health Science, Faculty of Veterinary Medicine, the Centre for Islamic Studies (C.I.S.), the *Cibiyar Nazarin Hausa* (Centre for Hausa Studies) and the Department of Extension Services. The main campus, situated some twelve kilometres to the north of the city, now houses the Central Administration together with the Faculties of Agriculture; Arts and Islamic Studies; Education and Extension Services; Law; Science; Social Sciences and Administration; and the Sokoto Energy Research Centre. The movement began to the main campus in 1982.

Teaching in the University began on October 20, 1977, with an initial enrolment of ninety three undergraduate students for the degree of Bachelor of Arts, Bachelor of Science and Bachelor of Arts or Science in Education. The Teaching and Research staff strength then was thirty three. In January 1978, one hundred and two students enrolled for a two-year pre-degree studies programme in the Humanities and Basic Sciences. With this modest start, the University formally graduated a total number of seventy-two students at the convocation held on 22nd November, 1980.

The University has witnessed significant growth since then, despite shortage of physical facilities. Thus, by the 1991/92 session, total students enrolment had risen to 4,920, while Teaching and Research strength stood at 329. Also at the end of 1990/91 session, the University graduated a total of 793 undergraduates, 7 masters, and 5 Doctor of Philosophy (Ph.D) graduates.

The University is now set to consolidate and stabilise its growth, with a target students enrolment of 8,000 in the next few years and a 60:40 ratio in favour of Science and Science-based disciplines.

3.2 FUNCTIONS OF THE ESTABLISHMENT DIVISION

The establishment division of the registry department is charge with the responsibility of handling all aspects of staff matters. The division is further divided into senior staff and junior staff establishments. The function of the establishment division is derived from the overall objective of the University thus: "As a centre of learning, the Usmanu Danfodiyo University would promote and disseminate knowledge, pursue research, contribute to the Universal truths and maintain the international character of a University". It would specifically:

- (i) Endeavour to identify the needs and aspiration of the people where it is located.
- (ii) Uphold and developed the ideals of the community amongst which it is situated and at the same time promote national unity.
- (iii) Cherish and Preserve the cultural heritage of the society; and
- (iv) Establish a two-way report between the university and the society.

To achieve the above objectives, a competent and dedicated staffing support is required. It is therefore pertinent that up to date record is maintained for these staff. This can only be achieved through a computerised staff records.

The following are the staff related function performed by the establishment division.

- (a) Recruitment of staff
- (b) Training of staff
- (c) Promotion of staff
- (d) Confirmation of staff appointment
- (e) Discipline of staff
- (f) Interpretation of circulars and pronouncements on staff issues.
- (g) Executing decisions of committees such as SSEC (Senior Staff Establishment Committee) ASAPC (Administrative and Services Appointment and Promotion Committee).
- (h) Representing the Registrar on faculties appointment communities.
- (i) Any other staff related matters as may be referred to in by the Registrar.

3.3 PROCEDURES OF THE MANUAL SYSTEM

The following are the procedures adopted by the establishment division in executing some of the functions enumerated in 3.1 above.

3.3.1 Staff Recruitment

This is usually done either by promoting staff within or by external advertisement. Applications are received from suitably qualified candidates either in response to earlier advertisement or unsolicited application. Upon the receipt of applications, the applications

are recorded in the application register which shows the following:

- (i) name of applicant
- (ii) qualification
- (iii) post applied for
- (iv) date of receipt of application
- (v) action taken by the establishment.

All registered applications are sent to the department/faculty conceived for comments and confirmation of vacancy. If there is vacancy, the establishment division invites the applicant(s) to an interview. Successful applicant(s) are sent letter(s) of provisional offer. Usually before the applicant(s) are sent letter(s) of provisional offer. Usually before the applicant(s) assume duty, her/she has to write a letter of acceptance of offer to the Registrar and should submit along with at least two referee's report. On assumption of duty, the applicant(s) completes documentation forms namely; Assumption of duty form and Personal data form.

- (i) **Assumption of duty form:** The staff on assumption of duty will collect this form from the establishment division and completes the form and he/she takes it to the dean/head of department of his/her department for signature. The form contains the information of staff name, date of assumption of duty and name of the department
- (ii) **Personal data form:** This contains all the personal information of the staff. It contains the following information ie surname and other names, date of birth, marital status, nationality, permanent domicile, academic qualification, next of kin, name of wife and children.

Four copies each of the two stated forms are expected to be completed by the staff. A copy of each forms are sent to the bursary department for salary documentation, a copy of each forms to his/her department, a copy of each forms to the Registry and a copy of each forms are to remain in the establishment division.

Apart from the above forms the establishment division maintains the following registers.

- (i) **Big book:** This is a register where all the names of staff who were given appointment are listed whether they accept or reject the offer. It contains the following information: date of offer, file no, name of staff, designation, faculty/department.
- (ii) **New Arrival Book:** This register contains list of only those that assume duty. It contains information like: Faculty/Department, Position, Date of assumption of duty, Point of entry, Qualification on entry, Institutions attended, Nature of offer, Duration (if on contract), Nationality, State of origin.

After all these processes (documentation) has been done, then he/she has been considered as a staff of the Usmanu Danfodiyo University either on tenure or contract basis.

3.3.2. Staff Promotion

Depending on the category of staff i.e. (academic or non-academic), the waiting period between date of employment as last promotion to another one ranges between two to three years.

As for graduate assistants they span a period of one year and then promoted to Assistant-Lecturers.

The promotion exercise commences with the establishment division dispatching promotion evaluation forms to various departments. All staff whether due or not due for promotion are to complete the form.

Completed forms are returned to the establishment division which will in turn pass the forms to the promotion and appointment committee. Promotion of staff between USS 6-11 is conclusive by the appointment and promotion committee whereas promotion of staff above USS 11 is referred to the council for final approval. It is the responsibility of establishment division to send letters of promotion to individual staff promoted.

3.3.3. Confirmation of Appointment

Confirmation of appointment is only applicable to staff on tenure appointment. The waiting period is usually two years from the date of appointment. The staff concerned will write an application, for confirmation of appointment to the Registrar through his/her head of department. The head of department will make his/her recommendation for/or against the confirmation. In the absence of any negative opinion by the head of department, the establishment division will send a letter of confirmation of appointment to the concerned staff.

3.4 TYPES OF FILE: There are three types of file record keeping; namely:

1. Subject file or committee file
2. Personal file
3. Document file

1. Subject or Committee file: These are file: These are files that treat matters concerning various committees which includes SSEC subcommittee, Disciplinary Committee, ASAP committee and Evaluation and Promotion Committee.

2. **Personal file:** This is a file of all categories of staff each contain the employees personal documents and referees report.

These are classified into divisions or categories of registry which includes; Bursary, Faculty office, Works and Services, etc.

3. **Document file:** This is a file of employee whose services are no longer with the university. It is always difficult to trace these kind of files due to their mixed up.

3.5 **PROBLEMS IDENTIFIED IN THE PRESENT SYSTEM**

- (a) Over reliance on human efforts which in most cases is not very reliable.
- (b) Slow speed of information generation.
- (c) Information on forms can be easily removed or up dated illegally to suit the staff.
- (d) Enormous time is wasted in recording and processing data.
- (e) Due to movement of staff files from one office to another, information contained in the file can be easily lost.
- (f) File can be easily lost.
- (h) Frequent staff changes in the establishment division could lead to lack of continuity in information recording and processing.
- (j) Too much stationeries are being wasted.

CHAPTER FOUR

ANALYSIS OF THE NEW SYSTEM

4.1 DESIGN OF DATABASE FILES

Fifteen database files were created for the successful running of the program. These are detailed below:

- (i) **STAF.DBF**: This file contains staff personal details like employee number, surname, other names, rank, date of birth, date assumed duty etc.
- (ii) **PUBLICAT.DBF**: This file contains all the publications written by the staff. It includes fields like employee number, publication title, place of publication and date of publication.
- (iii) **CONFEREN.DBF**: This file contains all the conferences/workshops attended by staff, the field names include employee number, conference title, place of conference, date of commencement and date ended conference and a logical field whether a staff has presented paper or not.
- (iv) **QUALIFIC.DBF**: This database file contains all the staff qualifications obtained. The field includes employee number, qualification title, place obtained qualification and the date obtained.
- (v) **COMMUNIC.DBF**: This file contains all the staff correspondences received within and outside the university. It contains field names like employee number, subject, reference number, date and a memo field.
- (vi) **W-EXPERI.DBF**: It has all the staff working experience and has the fields like employee number, rank, place, effective date and date ended.
- (vii) **OTHER-AP.DBF**: This dBASE file contains all the staff other appointments holding in the university. These appointments include Head of

department, Dean, committee membership, etc. It has field names like employee number, appointment title, date of commencement and ending of appointment.

- (viii) **LEAVE.DBF**: This is the type file that has all the staff leave eg casual, annual, sick etc. It has field names like, employee number, type of leave, place of leave, date of commencement of leave, date of resumption and date returned from leave.
- (ix) **WIVES.DBF**: This DBF file contains all the list of staff wives. It has the field names like, employee number, surname, othernames and date of birth.
- (x) **CHILDREN.DBF**: Children file contains all the list of staff children. It has the field names like, employee number, surname, othernames and date of birth.
- (xi) **CONFIRM.DBF**: The CONFIRM.DBF contains all the confirmed staff in the university. It two field names only ie employee number and date of confirmation.
- (xii) **SECURITY.DBF**: This file contains all the authorised users of the system for the systems security. It contains the field names like user name, user password and level of access.
- (xiii) **DEPARTME.DBF**: This file contains all the departments names in the university. It has two field names only ie, name of the department and its corresponding code.
- (xiv) **POPFIELD.DBF**: This file contains an expression that is to be displayed on the screen. It has only one field name that is description of the message.
- (xv) **TEMP.DBF**: This file is used for copying a particular structure which is to be copied for the display of the contents of that particular item. It has only one field name called experi.

4.2 OPERATIONAL PROCEDURE

The program has been compiled to executable one with the name STAF.EXE. When loaded it first takes the user into an interactive window waiting for the user to enter the username and the password for security reasons. Correct username and userpassword will automatically take the user to the main menu of the program else it will give the user two more chances to enter the correct variables and trials exhausted it will take you out from the dialogue window. The main menu is shown in the following table below:

Table 4.2

MAIN MENU
a. Enter Staff Information
b. View Staff Information
c. Modify Staff Information
d. Delete Staff Information
e. Print Reports
f. System Maintenance
EXIT

- a. **Enter Staff Information:** This menu allows the user to enter all the staff information eg new staff documentation, processing of staff leave, entering of text like memo etc.
- b. **View Staff Information:** The view menu allow the user to view the staff information that include things like staff details, staff curriculum vitae (CV), staff appointments etc.

- c. **Modify Staff Information:** Modify Staff Information menu is use for the modification of any information entered by user wrongly or otherwise and will effect the correction if the user permits the system by responding YES to effect the changes made.

- d. **Delete Staff Information:** This menu is use to delete any staff information or a staff completely from the system.

- e. **Print Reports:** This is where all the reports of staff information is being processed for printing. Things like staff CV, list of staff by departments, list of staff by state, list of all confirmed staff etc are obtained from this menu.

- f. **System Maintenance:** This menu is the one controlling the system's security. It allows the system's SUPERVISOR to add additional users, change the user's password, delete users from the system and view all the users names on the screen.

EXIT: This takes the user out of the system when the user wants to quit from the system.

CHAPTER FIVE

EXPERIMENTATION AND CONCLUSION

5.1 EXPERIMENTATION

Having successfully written the program and tested, the system has been demonstrated and experimented in all the menus designed, has been found working effectively and efficiently. The hardcopy of the source code is hereby attached as appendix.

5.2 CONCLUSION

It has been highly examined that, the **PERSONNEL RECORD SYSTEM** is more efficient, reliable, faster in terms of record keeping and easier to use than the manual record keeping. It has been found that so many things can be printed, modified, viewed, entered and deleted so easily.

APPENDIX

set talk off
set echo off
set safe off
set bell off
set exact on
set cent on
set scor off
set stat off
clear

CLOSE DATA

USE STAF IN 1

INDEX ON STAF_NO TAG BYSTAF_NO

USE OTHER_AP IN 2

SELE 2

INDEX ON STAF_NO TAG BYSTAF_NO

USE PUBLICAT IN 3

SELE 3

INDEX ON STAF_NO TAG BYSTAF_NO

DEFINE WINDOW XY_WAIT FROM 20,11 TO 23,65 COLOR N/BG,N/BG

msg = 'Select option and press ENTER'

DEFINE POPUP main_pop FROM 0,0 MESSAGE msg

DEFINE BAR 1 OF main_pop PROMPT " MAIN MENU" SKIP

DEFINE BAR 2 OF main_pop PROMPT REPL("=",25) SKIP

DEFINE BAR 3 OF main_pop PROMPT "Enter Staff Information >"

DEFINE BAR 4 OF main_pop PROMPT "View Staff Information >"

DEFINE BAR 5 OF main_pop PROMPT "Modify Staff Information>"


```

DEFINE BAR 6 OF main_pop PROMPT "Delete Staff Information>"
DEFINE BAR 7 OF main_pop PROMPT "Generate Report    >"
DEFINE BAR 8 OF main_pop PROMPT REPL("=",25) SKIP
DEFINE BAR 9 OF main_pop PROMPT "EXIT"
ON SELECTION POPUP main_pop DO main_menu

```

```

DEFINE POPUP entry_pop FROM 3,15 MESSAGE msg
DEFINE BAR 1 OF entry_pop PROMPT "STAFF DATA ENTRY MENU" SKIP
DEFINE BAR 2 OF entry_pop PROMPT REPL("=",25) SKIP
DEFINE BAR 3 OF entry_pop PROMPT "New Staff Documentation "
DEFINE BAR 4 OF entry_pop PROMPT "Promote/Demote Staff  "
DEFINE BAR 5 OF entry_pop PROMPT "Other Appoinments   "
DEFINE BAR 6 OF entry_pop PROMPT "Add Communications   "
DEFINE BAR 7 OF entry_pop PROMPT "Add Qualifications   "
DEFINE BAR 8 OF entry_pop PROMPT "Add Publications     "
DEFINE BAR 9 OF entry_pop PROMPT "Add Confernces/Workshop"
DEFINE BAR 10 OF entry_pop PROMPT "Process Leave        "
DEFINE BAR 11 OF entry_pop PROMPT "Add New Wives        "
DEFINE BAR 12 OF entry_pop PROMPT "Add New Children     "
DEFINE BAR 13 OF entry_pop PROMPT "Confirm Staff        "
DEFINE BAR 14 OF entry_pop PROMPT "Terminate Staff      "
DEFINE BAR 15 OF entry_pop PROMPT REPL("=",25) SKIP
DEFINE BAR 16 OF entry_pop PROMPT "Exit To Main Menu   "
ON SELECTION POPUP entry_pop DO entry_menu
DEFINE POPUP view_pop FROM 2,15 MESSAGE msg
DEFINE BAR 1 OF view_pop PROMPT "VIEW STAFF INFO MENU" SKIP
DEFINE BAR 2 OF view_pop PROMPT REPL("=",25) SKIP
DEFINE BAR 3 OF view_pop PROMPT "Personal Details"
DEFINE BAR 4 OF view_pop PROMPT "Working Experience"
DEFINE BAR 5 OF view_pop PROMPT "Qualifications"
DEFINE BAR 6 OF view_pop PROMPT "Publicatons"

```

```

DEFINE BAR 7 OF view_pop PROMPT "Conferences/Workshop"
DEFINE BAR 8 OF view_pop PROMPT "List of Wives"
DEFINE BAR 9 OF view_pop PROMPT "List of Children"
DEFINE BAR 10 OF view_pop PROMPT "Current Appointment"
DEFINE BAR 11 OF view_pop PROMPT "Communications"
DEFINE BAR 12 OF view_pop PROMPT "Staff CV"
DEFINE BAR 13 OF view_pop PROMPT "Leave History"
DEFINE BAR 14 OF view_pop PROMPT "List of Staff      >"
DEFINE BAR 15 OF view_pop PROMPT "Number of Staff  >"
DEFINE BAR 16 OF view_pop PROMPT REPL("=",25) SKIP
DEFINE BAR 17 OF view_pop PROMPT "Exit To Main Menu"
ON SELECTION POPUP view_pop DO view_menu

```

```

DEFINE POPUP modi_pop FROM 5,15 MESSAGE msg
DEFINE BAR 1 OF modi_pop PROMPT " STAFF MODIFY MENU" SKIP
DEFINE BAR 2 OF modi_pop PROMPT REPL("=",22) SKIP
DEFINE BAR 3 OF modi_pop PROMPT "Personal Detials"
DEFINE BAR 4 OF modi_pop PROMPT "Publications"
DEFINE BAR 5 OF modi_pop PROMPT "Qualifications"
DEFINE BAR 6 OF modi_pop PROMPT "Conferences/Workshop"
DEFINE BAR 7 OF modi_pop PROMPT "Wives"
DEFINE BAR 8 OF modi_pop PROMPT "Children"
DEFINE BAR 9 OF modi_pop PROMPT "Other Appointments"
DEFINE BAR 10 OF modi_pop PROMPT "Communications"
DEFINE BAR 11 OF modi_pop PROMPT "Leave"
DEFINE BAR 12 OF modi_pop PROMPT "Working Experience"
DEFINE BAR 13 OF modi_pop PROMPT "Confirmed Staff"
DEFINE BAR 14 OF modi_pop PROMPT "Terminated Staff"
DEFINE BAR 15 OF modi_pop PROMPT REPL("=",22) SKIP
DEFINE BAR 16 OF modi_pop PROMPT "Exit To Main Menu"
ON SELECTION POPUP modi_pop DO modi_menu

```

```
DEFINE POPUP delete_pop FROM 4,15 MESSAGE msg
DEFINE BAR 1 OF delete_pop PROMPT "  DELETE MENU" SKIP
DEFINE BAR 2 OF delete_pop PROMPT REPL("=",22) SKIP
DEFINE BAR 3 OF delete_pop PROMPT "Staff Information"
DEFINE BAR 4 OF delete_pop PROMPT "Publications"
DEFINE BAR 5 OF delete_pop PROMPT "Appointments"
DEFINE BAR 6 OF delete_pop PROMPT "Qualifications"
DEFINE BAR 7 OF delete_pop PROMPT "Conferences/Workshop"
DEFINE BAR 8 OF delete_pop PROMPT "Wives"
DEFINE BAR 9 OF delete_pop PROMPT "Children"
DEFINE BAR 10 OF delete_pop PROMPT "Communications"
DEFINE BAR 11 OF delete_pop PROMPT "Leave"
DEFINE BAR 12 OF delete_pop PROMPT "Working Experience"
DEFINE BAR 13 OF delete_pop PROMPT REPL("=",22) SKIP
DEFINE BAR 14 OF delete_pop PROMPT "Exit To Main Menu"
ON SELECTION POPUP delete_pop DO del_menu
```

```
DEFINE POPUP print_pop FROM 2,15 MESSAGE msg
DEFINE BAR 1 OF print_pop PROMPT "  PRINT REPORT MENU" SKIP
DEFINE BAR 2 OF print_pop PROMPT REPL("=",24) SKIP
DEFINE BAR 3 OF print_pop PROMPT "Staff Personal Details"
DEFINE BAR 4 OF print_pop PROMPT "Staff CV"
DEFINE BAR 5 OF print_pop PROMPT "List of Staff    >"
DEFINE BAR 6 OF print_pop PROMPT "Communications on Staff"
DEFINE BAR 7 OF print_pop PROMPT "Leave History"
DEFINE BAR 8 OF print_pop PROMPT "Wives & Children"
DEFINE BAR 9 OF print_pop PROMPT "Working Experience"
DEFINE BAR 10 OF print_pop PROMPT "Qualifications"
DEFINE BAR 11 OF print_pop PROMPT "Publications"
DEFINE BAR 12 OF print_pop PROMPT "Conferences/Workshop"
DEFINE BAR 13 OF print_pop PROMPT "Current Appointments"
```

```
DEFINE BAR 14 OF print_pop PROMPT REPL("=",24) SKIP
DEFINE BAR 15 OF print_pop PROMPT "Exit To Main Menu"
ON SELECTION POPUP print_pop DO print_menu
```

```
DEFINE WINDOW w_comm FROM 11,3 TO 23,76 DOUBLE COLOR n/w,n/w
```

```
ACTI POPUP main_pop
```

```
clear
```

```
close data
```

```
set exact off
```

```
set bell on
```

```
set safe on
```

```
set scor on
```

```
set stat on
```

```
set cent off
```

```
set talk on
```

```
set echo on
```

```
*****END OF MAIN*****
```

```
PROC main_menu
```

```
DO CASE
```

```
  CASE bar() = 3
```

```
    acti popup entry_pop
```

```
  CASE bar() = 4
```

```
    acti popup view_pop
```

```
  CASE bar() = 5
```

```
    acti popup modi_pop
```

```
  CASE bar() = 6
```

```
    acti popup delete_pop
```

```
  CASE bar() = 7
```

```
    acti popup print_pop
CASE bar() = 9
    deact popup
ENDCASE
RETURN
PROC entry_menu
DO CASE
    CASE bar() = 3
        DO DOCUMENT
    CASE bar() = 4
        mstaf_no = space(7)
        msg = 'Press ENTER to add other_ap for this staf_no'
        stop = .f.
        do while .not. stop
            do getstaf_no with mstaf_no,msg,4
            stop = (lastkey() = 27)
            if (.not. isbl(mstaf_no)) .and. (.not. stop)
                DO PROMOTE WITH mstaf_no
            endif
        enddo
    CASE bar() = 5
        mstaf_no = space(7)
        msg = 'Press ENTER to add other_ap for this staf_no'
        stop = .f.
        do while .not. stop
            do getstaf_no with mstaf_no,msg,4
            stop = (lastkey() = 27)
            if (.not. isbl(mstaf_no)) .and. (.not. stop)
                DO OTHER_APT WITH mstaf_no
            endif
        enddo
```

```
CASE bar() = 6
  mstaf_no = space(7)
  msg = 'Press ENTER to add communications for this staff'
  stop = .f.
  do while .not. stop
    do getstaf_no with mstaf_no,msg,4
    stop = (lastkey() = 27)
    if (.not. isbl(mstaf_no)) .and. (.not. stop)
      DO COMMUNIC WITH mstaf_no
    endif
  enddo
CASE bar() = 7
  mstaf_no = space(7)
  msg = 'Press ENTER to add qualification for this staf_no'
  stop = .f.
  do while .not. stop
    do getstaf_no with mstaf_no,msg,5
    stop = (lastkey() = 27)
    if (.not. isbl(mstaf_no)) .and. (.not. stop)
      DO QUALIFICA WITH mstaf_no
    endif
  enddo
CASE bar() = 8
  mstaf_no = space(7)
  msg = 'Press ENTER to add publication for this staf_no'
  stop = .f.
  do while .not. stop
    do getstaf_no with mstaf_no,msg,6
    stop = (lastkey() = 27)
    if (.not. isbl(mstaf_no)) .and. (.not. stop)
      DO PUBLICA WITH mstaf_no
    endif
```

```
    enddo
CASE bar() = 9
    mstaf_no = space(7)
    msg = 'Press ENTER to add confrence attended for this staff'
    stop = .f.
    do while .not. stop
        do getstaf_no with mstaf_no,msg,4
            stop = (lastkey() = 27)
            if (.not. isbl(mstaf_no)) .and. (.not. stop)
                DO CONFERENCE WITH mstaf_no
            endif
        enddo
    enddo
CASE bar() = 10
    mstaf_no = space(7)
    msg = 'Press ENTER to process leave for this staff'
    stop = .f.
    do while .not. stop
        do getstaf_no with mstaf_no,msg,4
            stop = (lastkey() = 27)
            if (.not. isbl(mstaf_no)) .and. (.not. stop)
                DO LEAVE WITH mstaf_no
            endif
        enddo
    enddo
CASE bar() = 11
    mstaf_no = space(7)
    msg = 'Press ENTER to add new wife for this staf_no'
    stop = .f.
    do while .not. stop
        do getstaf_no with mstaf_no,msg,3
            stop = (lastkey() = 27)
            if (.not. isbl(mstaf_no)) .and. (.not. stop)
                DO ADD_WIVES WITH mstaf_no
```

```
endif
enddo
CASE bar() = 12
  mstaf_no = space(7)
  msg = 'Press ENTER to add new child for this staf_no'
  stop = .f.
  do while .not. stop
    do getstaf_no with mstaf_no,msg,5
    stop = (lastkey() = 27)
    if (.not. isbl(mstaf_no)) .and. (.not. stop)
      DO ADD_CHILD WITH mstaf_no
    endif
  enddo
CASE bar() = 13
  mstaf_no = space(7)
  msg = 'Press ENTER to confirm this staf_no'
  stop = .f.
  do while .not. stop
    do getstaf_no with mstaf_no,msg,4
    stop = (lastkey() = 27)
    if (.not. isbl(mstaf_no)) .and. (.not. stop)
      DO CONFIRM WITH mstaf_no
    endif
  enddo
CASE bar() = 14
  mstaf_no = space(7)
  msg = 'Press ENTER to terminate this staf_no'
  stop = .f.
  do while .not. stop
    do getstaf_no with mstaf_no,msg,4
    stop = (lastkey() = 27)
    if (.not. isbl(mstaf_no)) .and. (.not. stop)
```



```
        DO TERMINATE WITH mstaf_no
    endif
enddo
CASE bar() = 16
    deact popup
ENDCASE
RETURN
PROC DOCUMENT
private mstaf_no,msurname,mothname,mstaf_cat,msex,mrank,mdate_rank,msalary_sc,;
    mdept,map_type,mduration,mdate_ap,mdate_asum,mentry_qal,mres_adr,mphone,;
    mm_status,mn_wives,mn_child,mnationalt,mstate,mlga,mh_town,mbirt_date,;
    mbirth_pl,mparm_adr,mnext_kin,mrelation,madr_kin,mkin_phone,msg,yesadd
mstaf_no = space(7)
do get_number with mstaf_no
if .not. isbl(mstaf_no)
sele 1
seek mstaf_no
if found()
    do alert with 'Sorry,this number already exist'
else
    msurname = SPACE(13)
    mothname = SPACE(17)
    mstaf_cat = SPACE(2)
    msex = SPACE(1)
    mrank = SPACE(25)
    mdate_rank = { / / }
    msalary_sc = SPACE(9)
    mdept = SPACE(17)
    map_type = SPACE(10)
    mduration = 0
    mdate_ap = { / / }
```

```
mdate_asum = { / / }
mentry_qal = SPACE(17)
mres_adr = SPACE(41)
mphone = SPACE(11)
mm_status = SPACE(9)
mn_wives = 0
mn_child = 0
mnationalt = SPACE(11)
mstate = SPACE(10)
mlga = SPACE(11)
mh_town = SPACE(13)
mbirt_date = { / / }
mbirth_pl = SPACE(27)
mparm_adr = SPACE(45)
mnext_kin = SPACE(25)
madr_kin = SPACE(38)
mkin_phone = SPACE(11)
msg = "Sure to add New Staf Documentation ? (Y/N) "
yesadd = .n.

DO get_staf1 with mstaf_no,msurname,mothname,mstaf_cat,msex,mrank,;
mdate_rank,msalary_sc,mdept,map_type,mduration,mdate_ap,mdate_asum,;
mentry_qal,mres_adr,mphone
DO get_staf2 with mm_status,mn_wives,mn_child,mnationalt,mstate,mlga,;
mh_town,mbirt_date,mbirth_pl,mparm_adr,mnext_kin,madr_kin,mkin_phone

yesadd = (.not. isbl(mstaf_no))
if yesadd
  do stamp with yesadd,msg
if yesadd
  SELE 1
```

APPEND BLANK

```

REPLACE STAF_NO WITH mstaf_no,SURNAME WITH msurname,OTHNAME
WITH, mothname,STAF_CAT WITH mstaf_cat,SEX WITH msex,RANK WITH
mrank, DATE_RANK WITH mdate_rank,SALARY_SC WITH msalary_sc,DEPT
WITH, mdept,AP_TYPE WITH map_type,DURATION WITH mduration,DATE_AP
WITH, mdate_ap,DATE_ASUM WITH mdate_asum,ENTRY_QAL WITH mentry_qal,;
ON_LEAVE WITH .N.,RES_ADR WITH mres_adr,PHONE WITH mphone,;
M_STATUS WITH mm_status,N_WIVES WITH mn_wives,N_CHILD WITH
mn_child,
NATIONALT WITH mnationalt,STATE WITH mstate,LGA WITH mlga,H_TOWN
WITH,mh_town,BIRT_DATE WITH mbirt_date,BIRTH_PL WITH
mbirth_pl,PARAM_ADR, WITH mparam_adr,NEXT_KIN WITH mnext_kin,ADR_KIN
WITHmadr_kin, KIN_PHONE WITH mkin_phone,CONFIRM WITH .N.,TERMINATE
WITH .N.
endif
endif
endif
endif
RETURN

```

PROC get_number

```

para mstaf_no
define window w_number from 5,25 to 9,66 color w/n
acti wind w_number
@ 1,2 SAY "Enter number for New Staff:"
@ 1,30 GET mstaf_no PICTURE "@!"
read
deact wind w_number
RETURN

```

PROCEDURE alert

```

para msg
set escape off

```

```

define window w_alert from 18,8 to 22,len(msg)+14 double color w/n
activate window w_alert

@ 1,2 say msg
WAIT "
deactivate window w_alert

set escape on

return

PROC PROMOTE

para mstaf_no

private mrank,mdate_rank,mnew_rank,mnew_salry,mnew_date,yespromote

    mrank = space(23)
    yespromote = .f.
    mdate_rank = { / / }
    mnew_rank = space(23)
    mnew_salry = space(9)
    mnew_date = { / / }
    do get_detail with mnew_rank,mnew_salry,mnew_date
    sele 1
    seek mstaf_no
    mrank = RANK
    mdate_rank = DATE_RANK
    msalary_sc = SALARY_SC
    do stamp with yespromote,'Sure to promote/demote '+mstaf_no+' ? (Y/N) '
    if yespromote
        replace RANK WITH mnew_rank,DATE_RANK WITH mnew_date,SALARY_SC
        WITH, mnew_salry
        use w_experi in 4
        sele 4
        append blank
        replace STAF_NO WITH mstaf_no,POST_HELD WITH mrank,PLACE WITH;
            'USMANU DANFODIYO UNIVERSITY SOKOTO',DATE_FROM WITH

```

```

mdate_rank,;
    DATE_TO WITH mnew_date
    use in 4
endif
RETURN

```

```

PROC get_detail
para mnew_rank,mnew_salry,mnew_date
    define window w_promote from 8,13 to 16,65 color w/n
    acti wind w_promote
    @ 1,3 SAY "To what rank ? :'"
    @ 1,20 GET Mnew_rank PICTURE "@!'"
    @ 3,3 SAY "To what salary scale ? :'"
    @ 3,28 GET Mnew_salry PICTURE "@!'"
    @ 5,3 SAY "Promotion/Demotion Date:"
    @ 5,28 GET Mnew_date PICTURE "@!'"
    read
    deact wind w_promote

```

```
RETURN
```

```
PROC OTHER_APT
```

```
para mstaf_no
```

```
private map_descrp,mdate_ap,yesadd,msg
```

```
    map_descrp = SPACE(25)
```

```
    mdate_ap = { / / }
```

```
    yesadd = .n.
```

```
    msg = 'Do you wish to confirm other appointment added ? (Y/N)'
```

```
    do get_apoint with mstaf_no,map_descrp,mdate_ap
```

```
    if yesadd
```

```
        do stamp with yesadd,msg
```

```
        if yesadd
```

```
            SELE 2
```

APPEND BLANK

REPLACE STAF_NO WITH mstaf_no,AP_DESCRP WITH map_descrp,;

DATE_AP WITH mdate_ap

endif

endif

RETURN

PROC get_apoint

PARA mstaf_no,map_descrp,mdate_ap

define window w_apoint from 3,17 to 18,71 color w/n

acti window w_apoint

@ 2,9 TO 4,41

@ 3,11 SAY "STAFF OTHER APPOINTMENTS FORM"

@ 7,9 SAY "STAF_NO:"

@ 7,17 SAY Mstaf_no PICTURE "@!"

@ 9,9 SAY "AP_DESCRP:"

@ 9,19 GET Map_descrp PICTURE "@!"

@ 11,9 SAY "DATE_AP:"

@ 11,17 GET Mdate_ap

read

deact wind w_apoint

RETURN

PROC PUBLICA

para mstaf_no

private mpub_descrp,mpublisher,mdate_pub,yesadd,msg

mpub_descrp = space(37)

mpublisher = space(31)

mdate_pub = { / / }

yesadd = .n.

msg = 'Do you wish to confirm publication added ? (Y/N)'

do get_publica with mstaf_no,mpub_descrp,mpublisher,mdate_pub

```
if yesadd
  do stamp with yesadd,msg
  if yesadd
    SELE 3
    APPEND BLANK
    REPLACE STAF_NO WITH mstaf_no,PUB_DESCR WITH mpub_descr,;
    DATE_PUB WITH mdate_pub
  endif
endif
```

```
RETURN
```

PROC QUALIFICA

```
para mstaf_no
```

```
private mcert_obt,mplace_obt,mdate_from,mdate_to,yesadd,msg
```

```
  mcert_obt = space(38)
```

```
  mplace_obt = space(37)
```

```
  mdate_from = { / / }
```

```
  mdate_to = { / / }
```

```
  yesadd = .n.
```

```
  msg = 'Do you wish to confirm qualification added ? (Y/N) '
```

```
  do get_quali with mstaf_no,mcert_obt,mplace_obt,mdate_from,mdate_to
```

```
if yesadd
```

```
  do stamp with yesadd,msg
```

```
  if yesadd
```

```
    USE QUALIFIC IN 4
```

```
    SELE 4
```

```
    INDEX ON STAF_NO TAG BYSTAF_NO
```

```
    APPEND BLANK
```

```
    REPLACE STAF_NO WITH mstaf_no,CERT_OBT WITH mcert_obt,;
```

```
    PLACE_OBT WITH mplace_obt,DATE_FROM WITH mdate_from,DATE_TO;
```

```
    WITH mdate_to
```

```
        use in 4
    endif
endif
RETURN
```

```
PROC get_quali
PARA mstaf_no,mcert_obt,mplace_obt,mdate_from,mdate_to,msg
    define window w_quali FROM 2,13 TO 16,67 color w/n
    acti wind w_quali
    @ 1,18 TO 3,35
    @ 2,20 SAY "QUALIFICATIONS"
    @ 5,3 SAY "STAF_NO:"
    @ 5,11 SAY Mstaf_no PICTURE "@!"
    @ 7,3 SAY "CERT_OBT:"
    @ 7,12 GET Mcert_obt PICTURE "@!"
    @ 9,3 SAY "PLACE_OBT:"
    @ 9,13 GET Mplace_obt PICTURE "@!"
    @ 11,3 SAY "DATE_FROM:"
    @ 11,13 GET Mdate_from
    @ 11,26 SAY "DATE_TO:"
    @ 11,34 GET Mdate_to
    read
    deact wind w_quali
RETURN
```

```
PROC CONFERENCE
para mstaf_no
private mconf_desc,mpaper_pre,mtitle,mplace,mdate_from,mdate_to,yesadd,msg
    mconf_desc = space(30)
    mpaper_pre = .f.
    mtitle = space(35)
    mplace = space(25)
```



```

mdate_from = { / / }
mdate_to = { / / }
yesadd = .n.
msg = 'Do you wish to confirm conference added ? (Y/N) '
do get_conf with mstaf_no,mconf_desc,mpaper_pre,mtitle,mplace,;
                mdate_from,mdate_to

```

```

if yesadd

```

```

    do stamp with yesadd,msg

```

```

    if yesadd

```

```

        USE CONFEREN IN 4

```

```

        SELE 4

```

```

        INDEX ON STAF_NO TAG BYSTAF_NO

```

```

        APPEND BLANK

```

```

        REPLACE STAF_NO WITH mstaf_no,CONF_DESC WITH mconf_desc,;

```

```

        PAPER_PRE WITH mpaper_pre,TITLE WITH mtitle,PLACE WITH mplace,;

```

```

        DATE_FROM WITH mdate_from,DATE_TO WITH mdate_to

```

```

        use in 4

```

```

    endif

```

```

endif

```

```

RETURN

```

```

PROC get_conf

```

```

PARA mstaf_no,mconf_desc,mpaper_pre,mtitle,mplace,mdate_from,mdate_to

```

```

    define window w_conf FROM 3,13 TO 20,67 color w/n

```

```

    acti wind w_conf

```

```

    @ 0,19 TO 2,36

```

```

    @ 1,22 SAY "CONFERENCES"

```

```

    @ 4,3 SAY "STAF_NO:"

```

```

    @ 4,11 SAY Mstaf_no PICTURE "@!"

```

```

    @ 6,3 SAY "CONF_DESC:"

```

```

    @ 6,14 GET Mconf_desc PICTURE "@!"

```

```

@ 8,3 SAY "PAPER_PRE:"
@ 8,14 GET Mpaper_pre PICTURE "@!"
@ 10,3 SAY "TITLE:"
@ 10,10 GET Mtitle PICTURE "@!"
@ 12,3 SAY "PLACE:"
@ 12,10 GET Mplace PICTURE "@!"
@ 14,3 SAY "DATE_FROM:"
@ 14,14 GET Mdate_from
@ 14,26 SAY "DATE_TO:"
@ 14,35 GET Mdate_to
read
deact wind w_conf

```

RETURN

PROC ADD_WIVES

para mstaf_no

private mname,mbirt_date,yesadd,msg

mname = SPACE(25)

mbirt_date = { / / }

yesadd = .n.

msg = 'Sure to confirm New Wife added ? (Y/N)'

do get_wife with mstaf_no,mname,mbirt_date

yesadd = (.not. isbl(mstaf_no))

if yesadd

do stamp with yesadd,msg

if yesadd

USE WIVES IN 4

SELE 4

INDEX ON STAF_NO TAG BYSTAF_NO

APPEND BLANK

REPLACE STAF_NO WITH mstaf_no,NAME WITH mname,;

```
BIRT_DATE WITH mbirt_date
```

```
use in 4
```

```
endif
```

```
endif
```

```
RETURN
```

```
PROC get_wife
```

```
PARA mstaf_no,mname,mbirt_date
```

```
define window w_wife from 3,17 to 16,69 color w/n
```

```
acti window w_wife
```

```
@ 1,17 TO 3,29
```

```
@ 2,19 SAY "NEW WIVES"
```

```
@ 5,9 SAY "STAF_NO:"
```

```
@ 5,17 SAY Mstaf_no PICTURE "@!"
```

```
@ 7,9 SAY "NAME:"
```

```
@ 7,15 GET Mname PICTURE "@!"
```

```
@ 9,9 SAY "BIRT_DATE:"
```

```
@ 9,20 GET Mbirt_date
```

```
read
```

```
deact wind w_wife
```

```
RETURN
```

```
PROC ADD_CHILD
```

```
para mstaf_no
```

```
private mname,mbirt_date,yesadd,msg
```

```
mname = SPACE(25)
```

```
mbirt_date = { / / }
```

```
yesadd = .n.
```

```
msg = 'Do you wish to confirm New Child added ? (Y/N) '
```

```
do get_child with mstaf_no,mname,mbirt_date
```

```
yesadd = (.not. isbl(mstaf_no)).
```

```
if yesadd
  do stamp with yesadd,msg
  if yesadd
    USE CHILDREN IN 4
    SELE 4
    INDEX ON STAF_NO TAG BYSTAF_NO
    APPEND BLANK
    REPLACE STAF_NO WITH mstaf_no,NAME WITH mname,;
    BIRT_DATE WITH mbirt_date
    use in 4
  endif
endif
RETURN
```

```
PROC get_child
  PARA mstaf_no,mname,mbirt_date
  define window w_child from 3,17 to 16,69 color w/n
  acti window w_child
  @ 1,17 TO 3,29
  @ 2,19 SAY "NEW CHILD"
  @ 5,9 SAY "STAF_NO:"
  @ 5,17 SAY Mstaf_no PICTURE "@!"
  @ 7,9 SAY "NAME:"
  @ 7,15 GET Mname PICTURE "@!"
  @ 9,9 SAY "BIRT_DATE:"
  @ 9,20 GET Mbirt_date
  read
  deact wind w_child
RETURN
```

```
PROC CONFIRM
```

```
para mstaf_no
```

```
private mdate_conf,yesadd,msg
```

```
    mdate_conf = { / / }
```

```
    yesadd = .n.
```

```
    msg = 'Sure to confirm this staff appointment ? (Y/N) '
```

```
    do date_confm with mdate_conf
```

```
    yesadd = (.not. isbl(mstaf_no))
```

```
    if yesadd
```

```
        do stamp with yesadd,msg
```

```
        if yesadd
```

```
            SELE 1
```

```
            SEEK mstaf_no
```

```
            REPLACE CONFIRM WITH .Y.,DATE_CONFM WITH mdate_conf
```

```
        endif
```

```
    endif
```

```
RETURN
```

```
PROC date_confm
```

```
para mdate_confm
```

```
    define window d_confm from 15,42 to 19,68 color w/n
```

```
    acti wind d_confm
```

```
    @ 1,2 SAY "DATE_CONF:"
```

```
    @ 1,12 GET Mdate_conf
```

```
    read
```

```
    deact wind d_confm
```

```
RETURN
```

```
PROC TERMINATE
```

```
para mstaf_no
```

```
private mtype_term,mdate_term,yesterm,msg1,msg
```

```
    mtype_term = space(15)
```

```
mdate_term = { / / }
```

```
yesterm = .f.
```

```
msg1 = 'Sure to terminate appointment of '+mstaf_no+' ? (Y/N) '
```

```
msg = 'Select type ENTER to terminate appointment of '+mstaf_no
```

```
DEFINE POPUP term_pop FROM 14,42 MESSAGE msg
```

```
DEFINE BAR 1 OF term_pop PROMPT "Death"
```

```
DEFINE BAR 2 OF term_pop PROMPT "Resign"
```

```
DEFINE BAR 3 OF term_pop PROMPT "Retire"
```

```
DEFINE BAR 4 OF term_pop PROMPT "Transfer"
```

```
DEFINE BAR 5 OF term_pop PROMPT "Dismissal"
```

```
DEFINE BAR 6 OF term_pop PROMPT "Withdrawal"
```

```
DEFINE BAR 7 OF term_pop PROMPT "Other types"
```

```
ON SELECTION POPUP term_pop DO term_type with mtype_term,msg
```

```
ACTI POPUP term_pop
```

```
if (mtype_term = 'Death').or.;
```

```
  (mtype_term = 'Retire').or.;
```

```
  (mtype_term = 'Transfer').or.;
```

```
  (mtype_term = 'Dismissal').or.;
```

```
  (mtype_term = 'Withdrawal').or.;
```

```
  (mtype_term = 'Other types')
```

```
do date_term with mdate_term
```

```
do stamp with yesterm,msg1
```

```
if yesterm
```

```
  SELE 1
```

```
  SEEK mstaf_no
```

```
  REPLACE TERMINATE WITH .Y.,TYPE_TERM WITH mtype_term,;
```

```
  DATE_TERM WITH mdate_term
```

```
endif
```

```
endif
```

```
RETURN
```

```
PROC term_type
```

```
para mtype_term
```

```
  if lastkey() = 13
```

```
    mtype_term = prompt()
```

```
  endif
```

```
  deact popup
```

```
RETURN
```

```
PROC date_term
```

```
para mdate_term
```

```
  define window w_term from 15,42 to 19,68 color w/n
```

```
  acti wind w_term
```

```
  @ 1,2 SAY "DATE_TERM:"
```

```
  @ 1,12 GET Mdate_term
```

```
  read
```

```
  deact wind w_term
```

```
RETURN
```

```
PROC COMMUNIC
```

```
para mstaf_no
```

```
private mfrom,mref,msubject,mdate,enter_com
```

```
  mfrom = space(25)
```

```
  mref = space(15)
```

```
  msubject = space(35)
```

```
  mdate = { / / }
```

```
  enter_com = .f.
```

```
  msg = 'Sure to enter communication for '+mstaf_no+' ? (Y/N) '
```

```
  use communic in 4
```

```
  sele 4
```

```
  append blank
```

```
  do get_comm with mstaf_no,mfrom,mref,msubject,mdate
```

```
  replace STAF_NO WITH mstaf_no, FROM WITH mfrom, REF WITH mref, SUBJECT;
```

```

        WITH msubject,DATE WITH mdate
do stamp with enter_com,msg
if .not. enter_com
    go bottom
    delete
    pack
endif
use in 4
RETURN

PROC LEAVE
para mstaf_no
private to_return,mleave_typ,mdate_from,mdate_to,mplace,mdate_retn,yesleave,;
    yesreturn,msg1,msg2
to_return = ''
mleave_typ = '      '
mdate_from = { / / }
mdate_to = { / / }
mplace = space(31)
mdate_retn = { / / }
yesleave = .f.
yesreturn = .f.
msg1 = 'Select option and press ENTER to process leave for '+mstaf_no
msg2 = 'Sure to process leave for '+mstaf_no +' ? (Y/N) '

DEFINE POPUP leave_pop FROM 13,42 MESSAGE msg1
DEFINE BAR 1 OF leave_pop PROMPT "Going"
DEFINE BAR 2 OF leave_pop PROMPT "Resuming"
ON SELECTION POPUP leave_pop DO go_to with to_return

DEFINE POPUP going_pop FROM 13,42 MESSAGE msg1

```



```

DEFINE BAR 1 OF going_pop PROMPT "Casual leave"
DEFINE BAR 2 OF going_pop PROMPT "Annual leave"
DEFINE BAR 3 OF going_pop PROMPT "Servatical leave"
DEFINE BAR 4 OF going_pop PROMPT "Study leave"
DEFINE BAR 5 OF going_pop PROMPT "Other leaves"
ON SELECTION POPUP going_pop DO get_type with mleave_typ

```

```

ACTI POPUP leave_pop

```

```

if to_return = 'G'

```

```

    acti popup going_pop

```

```

    do get_leave with mleave_typ,mdate_from,mdate_to,mplace

```

```

    do stamp with yesleave,msg2

```

```

    if yesleave

```

```

        sele 1

```

```

        seek mstaf_no

```

```

        replace ON_LEAVE WITH .Y.

```

```

        use leave in 4

```

```

        sele 4

```

```

        append blank

```

```

replace STAF_NO WITH mstaf_no,LEAVE_TYP WITH mleave_typ,DATE_FROM;

```

```

WITH mdate_from,DATE_TO WITH mdate_to,PLACE WITH mplace

```

```

    endif

```

```

    use in 4

```

```

else

```

```

    do date_resum with mdate_retn

```

```

    do stamp with yesreturn,'Sure to process returning of '+mstaf_no+' ? (Y/N) '

```

```

    if yesreturn

```

```

        sele 1

```

```

        seek mstaf_no

```

```

        replace ON_LEAVE WITH .N.

```

```

        use leave in 4

```

```
sele 4
set filt to STAF_NO = mstaf_no
replace DATE_RETN WITH mdate_retn for isbl(DATE_RETN)
endif
use in 4
endif
RETURN
```

```
PROC go_to
para to_return
if lastkey() = 13
to_return = subs(prompt(),1,1)
endif
deact popup
RETURN
```

```
PROC get_type
para mleave_typ
if lastkey() = 13
mleave_typ = prompt()
endif
deact popup
RETURN
```

```
PROC get_leave
para mleave_typ,mdate_from,mdate_to,mplace
define window w_leave from 3,15 to 17,61 color w/n
acti wind w_leave
@ 1,16 SAY "LEAVE PROCESS"
@ 3,4 SAY "STAF_NO:"
@ 3,13 GET Mstaf_no PICTURE "@!"
@ 5,4 SAY "LEAVE_TYPE:"
```

```
@ 5,15 GET Mleave_typ PICTURE "@!"
```

```
@ 7,4 SAY "DATE_FROM:"
```

```
@ 7,14 GET Mdate_from
```

```
@ 9,4 SAY "DATE_TO:"
```

```
@ 9,12 GET Mdate_to
```

```
@ 11,4 SAY "PLACE:"
```

```
@ 11,10 GET Mplace PICTURE "@!"
```

```
read
```

```
deact wind w_leave
```

```
RETURN
```

```
PROC date_resum
```

```
para mdate_retn
```

```
define window w_retn from 15,30 to 19,59 color w/n
```

```
acti wind w_retn
```

```
@ 1,2 SAY "DATE_RETN:"
```

```
@ 1,13 GET Mdate_retn
```

```
read
```

```
deact wind w_retn
```

```
RETURN
```

```
PROC get_number
```

```
para mstaf_no
```

```
define window w_number from 5,25 to 9,66 color w/n
```

```
acti wind w_number
```

```
@ 1,2 SAY "Enter number for New Staff:"
```

```
@ 1,30 GET mstaf_no PICTURE "@!"
```

```
read
```

```
deact wind w_number
```

```
RETURN
```

```
PROC view_menu
DO CASE
  CASE bar() = 3
    DO vew_detail
  CASE bar() = 4
    DO vew_experi
  CASE bar() = 5
    DO vew_qualfi
  CASE bar() = 6
    DO vew_publi
  CASE bar() = 7
    DO vew_conf
  CASE bar() = 8
    DO vew_wives
  CASE bar() = 9
    DO vew_child
  CASE bar() = 10
    DO vew_appt
  CASE bar() = 11
    DO vew_comm
  CASE bar() = 12
    DO vew_stafcv
  CASE bar() = 13
    DO vew_leave
  CASE bar() = 14
    DO vew_list
  CASE bar() = 15
    DO vew_number
  CASE bar() = 17
    deact popup
ENDCASE
RETURN
```

PROC view_detail

```
private mstaf_no,msurname,mothname,mstaf_cat,msex,mrank,mdate_rank,msalary_sc,;
  mdept,map_type,mduration,mdate_ap,mdate_asum,mentry_qal,mres_adr,mphone,;
  mm_status,mn_wives,mn_child,mnationalt,mstate,mlga,mh_town,mbirt_date,;
  mbirth_pl,mparm_adr,mnext_kin,madr_kin,mkin_phone,msg,stop
mstaf_no = space(7)
msg ='Select number to view personal details of '+mstaf_no
stop = .f.
do while .not. stop
  DO getstaf_no with mstaf_no,msg,4
  stop = (lastkey() = 27)
  if (.not. isbl(mstaf_no)) .and. (.not. stop)
    sele 1
    if seek(mstaf_no)
      msurname = SURNAME
      mothname = OTHNAME
      mstaf_cat = STAF_CAT
      msex = SEX
      mrank = RANK
      mdate_rank = DATE_RANK
      msalary_sc = SALARY_SC
      mdept = DEPT
      map_type = AP_TYPE
      mduration = DURATION
      mdate_ap = DATE_AP
      mdate_asum = DATE_ASUM
      mentry_qal = ENTRY_QAL
      mres_adr = RES_ADR
      mphone = PHONE
      mm_status = M_STATUS
      mn_wives = N_WIVES
      mn_child = N_CHILD
```

mnationalt = NATIONALT

mstate = STATE

mlga = LGA

mh_town = H_TOWN

mbirt_date = BIRT_DATE

mbirth_pl = BIRTH_PL

mparm_adr = PARM_ADR

mnext_kin = NEXT_KIN

madr_kin = ADR_KIN

mkina_phone = KIN_PHONE

DO get_data1 with mstaf_no,msurname,mothname,mstaf_cat,msex,mrank,;
 mdate_rank,msalary_sc,mdept,map_type,mduration,mdate_ap,mdate_asum,;
 mentry_qal,mres_adr,mphone

DO get_data2 with mm_status,mn_wives,mn_child,mnationalt,mstate,mlga,;
 mh_town,mbirt_date,mbirth_pl,mparm_adr,mnext_kin,madr_kin,mkina_phone

endif

endif

enddo

RETURN

PROC get_data1

PARA mstaf_no,msurname,mothname,mstaf_cat,msex,mrank,mdate_rank,msalary_sc,;
 mdept,map_type,mduration,mdate_ap,mdate_asum,mentry_qal,mres_adr,mphone

define window w_data1 from 0,0 to 21,79 color w/n

acti wind w_data1

@ 0,25 TO 2,51

@ 1,27 SAY "STAFF PERSONAL DETAILS" color rg+/gb+

@ 4,2 SAY "STAF_NO:"

@ 4,10 SAY mstaf_no PICTURE "@!" color rg+/gb+

@ 4,21 SAY "SURNAME:"

@ 4,30 SAY msurname PICTURE "@!" color rg+/gb+
@ 4,47 SAY "OTHNAME:"
@ 4,55 SAY mothname PICTURE "@!" color rg+/gb+
@ 7,2 SAY "STAF_CAT:"
@ 7,12 SAY mstaf_cat PICTURE "!!" color rg+/gb+
@ 7,19 SAY "SEX:"
@ 7,24 SAY msex PICTURE "!" color rg+/gb+
@ 7,29 SAY "RANK:"
@ 7,35 SAY mrank PICTURE "@!" color rg+/gb+
@ 10,2 SAY "DATE_RANK:"
@ 10,13 SAY mdate_rank color rg+/gb+
@ 10,27 SAY "SALARY_SC:"
@ 10,38 SAY msalary_sc PICTURE "@!" color rg+/gb+
@ 10,51 SAY "DEPT:"
@ 10,57 SAY mdept PICTURE "@!" color rg+/gb+
@ 13,2 SAY "AP_TYPE:"
@ 13,11 SAY map_type PICTURE "@!" color rg+/gb+
@ 13,26 SAY "DURATION:"
@ 13,35 SAY mduration PICTURE "99" color rg+/gb+
@ 13,42 SAY "DATE_AP:"
@ 13,51 SAY mdate_ap color rg+/gb+
@ 16,2 SAY "DATE_ASUM:"
@ 16,13 SAY mdate_asum color rg+/gb+
@ 16,27 SAY "ENTRY_QAL:"
@ 16,38 SAY mentry_qal PICTURE "@!" color rg+/gb+
@ 18,2 SAY "RES_ADR:"
@ 18,11 SAY mres_adr PICTURE "@!" color rg+/gb+
@ 18,56 SAY "PHONE:"
@ 18,63 SAY mphone PICTURE "@!" color rg+/gb+
read
deact wind w_data1

RETURN

REFERENCES

1. Advanced Microsoft Visual Basic, second edition, Michael Halvorson, p. 15
3. Bolrland dbase for DOS 5.0, Programmers Guide, pp. 34-37
4. Bolrland dbase for DOS 5.0, Users Guide, p. 19
5. Bolrland dbase for DOS 5.0, Language Reference
6. Hitchhiker's Guide to Visual Basic and SQL Server, sixth edition, pp. 23-27
7. Introduction to Basic Programming, M.C Garry, 1980, p. 31
8. Introduction to Computer Systems, Lecture notes by Mr. Badmos, 1995, p. 41
9. Microsoft Visual Basic 6.0, Developers Workshop, Michael Halvorson, p. 35
10. Microsoft Visual Basic 6.0, Learn Visual J++ Now, Michael Halvorson, pp. 13-15
11. Microsoft Visual Basic 6.0 Programmers Guide, Michael Halvorson, pp. 35-39
12. Microsoft Visual Basic 6.0 Reference Library, Michael Halvorson, p. 65
13. Oxford Advanced Learner's Dictionary, A.S Hornby, fifth edition, oxford university press, p. 15
14. Visual Basic 3.0 for Windows Handbook, Gary Cornell, Osborne McGraw Hill, USA.