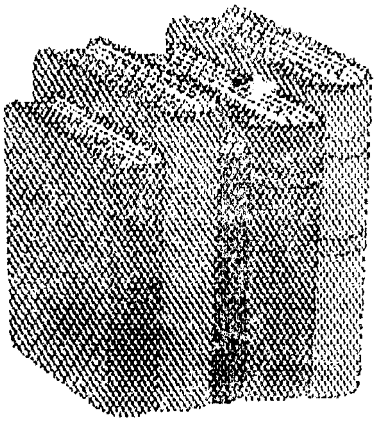
LIBRARY MANAGEMENT SYSTEM



Case Study By

Hariharan. S, Jitesh Menon. P & Kannan. N

October 1994

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE - 641 006

96CC06 96IT17

Department of

....

....

LABORATORY RECORD

Name .... Roll No. ....

Class .... Branch ....

Certified Bonafide Record of Work Done by

Mr. ....

Place ....

Staff in Charge

Date ....

University Register No. ....

Submitted for the University Practical Examination

Held on ....

Internal Examiner

External Examiner

CONTENTS

SYNOPSIS:

Library management is a separate branch of science dealing with the classification of book, journals, papers and their classification. Even today, researchers and librarians continue to work on newer and efficient methods of management. The enormous size in the number of books and the tedious task of maintenance and classification has always been a great challenge.

The task of classification, maintaining a set of databases of the member names, names of books, updating the database....etc. always results in a large number of records and files. Searching through the records and classifying them is not an easy task. Moreover, as the number of books increases the volume of paper material needed to maintain them increases proportionately. Now with the advent of computers, this task has been made all the more easier. Computers work at an extremely faster rate when compared to humans. The amount of paper work and paper material is almost nil.

In this case study, we have basically simplified the process of library management by computerizing the entire system. Simultaneously we have provided new facilities, basic enough to be understood by any librarian who has absolutely no knowledge of computers. To achieve this we adopted Microsoft FoxPro -- a highly advanced database management system. This way, the users have the advantage of saving time, money and space. The result -- an overall increase in the efficiency of library management.

facilities of instantaneous access to databases and answers to simple queries in no more than few seconds.

1.3 Problem definition

The problem could be basically defined as an intricate classification of books based on various fields and maintenance of a database based on various records. The definition could be further attenuated as the simultaneous updating, classification and maintenance of books with an optimal saving in time and physical space.

1.4 Solution details

The solution to the above mentioned problem primarily involves the use of a database management software that contains features necessary for basic programming. Using various databases essentially eases the entire understanding of the problem as a whole. This way updating or increasing the size of the databases will become easier and simpler.

Simple basic algorithms facilitate quicker searching techniques and classification methods. Minimal opening of files should be ensured to save loading and response time. Relational key hierarchy should be used to as far as possible to link databases, since relational databases are not supported by this version of FoxPro.

This method involves efficient access and improvised enhancement techniques. A more interactive method of user input is intended. As far as possible commands should be avoided and key driven interactive user interfaces created. We assume thc real time user has absolutely no knowledge of a computer.

2.0 DESIGN OVERVIEW

2.1 Introduction

An overview of the design involves primarily the creation of databases and modules to maintain these databases. Our basic design involves using extensive pop-up menus supported by FoxPro that provide better user interaction. The rest of the modules contain data entry or data display by the computer. Since, this system was being carried out for our college library, we had to be all the more careful with interface design and dialogs. The librarian who had to use it obviously had no idea of a computer. We have tried to provide on line help for every part of the system.

The usage of a single relational key has been maintained throughout. The use of a relational key facilitates access to other database, to which we need a link. the relational hierarchy of databases is not fully available in FoxPro. Thus updating a single database does not simultaneously update the other.

The system has been designed in such a way that it supports further modification or addition. Extensive usage's of simple codes and a large number of smaller modules facilitate easier understanding of the code. Thus, any further improvement in the code may be carried out with ease.

2.2 Issues involved in design

The design process was not a walk through. We had to go through many hurdles before we could arrive at any particular optimal solution.

For example, in the borrowing module -- the system had to check whether there existed such a member borrowing that book. For this it had to use the member database. Secondly, it had to check whether the book with the given accession number was present. This it did by checking the name database. Thirdly, it checks to see if the book is available for borrowing. Fourthly, it checks whether the member has got a free card and then all the databases are used once again to update after borrowing. This process involves a lot of searching. So, one of the main issue was speed that went sluggish when large databases were used. The similar problem was faced while handling the lending section.

Other problems could involve hardware constraints. The system generally requires a large secondary memory for optimal storage and a sufficiently fast processor for quicker access.

2.3 Design Details

We restricted the number of databases, as far as possible, to the minimum. This way we could reduce access time and understanding was more lucid. Although, we could not compromise for the large amount of searching done while browsing through the fields of these large databases.

Databases used:

- Member database.
- Book's database.
- Circulation database

Book database – NAME.dbf

NAME	TYPE	WIDTH	DECIMAL	DESCRIPTION
BOOK_NAME	Character	75	0	Name of the book
AUTHOR	Character	30	0	Author Name
VOLUME	Numeric	3	0	Volume of book
EDITION	Character	10	0	Edition of book
SUBJECT	Character	15	0	Subject of book
COPIES	Numeric	2	0	Copies available
T_OR_REF	Character	1	0	Text/Reference Book
BNK_OR_GNR	Character	1	0	Book bank/General
CALL_NO	Character	10	0	Call Number
ACCN_NO	Numeric	10	0	Accession number
STATUS	Character	1	0	Book available
PUBLISHER	Character	30	0	Publisher name
PRICE	Numeric	7	2	Price of book
DATE	Date	8	0	Date Purchased
AGENTS	Character	30	0	Agents for books
BILL_NO	Numeric	6	0	Bill Number

Member database – MEMBER.dbf

NAME	TYPE	WIDTH	DECIMAL	DESCRIPTION
NAME	Character	25	0	Name of the member
CARD_NO	Numeric	5	0	Card Number of member
ROLL_NO	Character	7	0	Roll Number of member
F_OR_S	Character	7	0	Staff/Student
MAX_BKS	Numeric	1	0	Maximum Books
AC_BNK	Character	1	0	Access to Book bank

Circulation database – CIRC.dbf

NAME	TYPE	WIDTH	DECIMAL	DESCRIPTION
ROLL_NO	Character	7	0	Roll No. of the member
TOT_BOOK	Numeric	1	0	Total Books allowed
ACCN_NO_1	Numeric	1	0	Accession no. of book 1
ACCN_NO_2	Numeric	1	0	Accession no. of book 2
ACCN_NO_3	Numeric	1	0	Accession no. of book 3
ACCN_NO_4	Numeric	1	0	Accession no. of book 4
ACCN_NO_5	Numeric	1	0	Accession no. of book 5
DUE_DATE_1	Date	8	0	Due date of Book 1
DUE_DATE_2	Date	8	0	Due date of Book 2
DUE_DATE_3	Date	8	0	Due date of Book 3
DUE_DATE_4	Date	8	0	Due date of Book 4
DUE_DATE_5	Date	8	0	Due date of Book 5
FINE	Numeric	5	2	Fine imposed

- *Book Database:*

The book database primarily deals with storage of books and their related fields. The fields are as shown above. Basic fields which are of importance are included.

- *Member Database:*

The member database deals with the storage of member names and their allotment details. The fields are as shown above. Specific details maximum book allowed are of prime importance.

- *Circulation Database:*

The circulation database contains details regarding circulation. It's nothing but a form of book card available in the library.

3.0 IMPLEMENTATION DETAILS

3.1 Introduction

The entire case study has been implemented using the world famous database management software -- Microsoft FoxPro. The FoxPro software was developed by the Microsoft corporation and is basically intended to abet database program and develop custom built database software.

The FoxPro program is a sophisticated piece of a software and has very easy to use interface. It only takes a matter of few minutes to have a basic understanding of the software. However, its the programming capability of FoxPro that makes it an outstanding database management system. A few simple commands form the basic tools for good programming. A primary reason for our choosing FoxPro as the software for our case study was its flexibility and ease of use with which we could produce the desired result.

The minimal hardware requirement is an XT computer with at least 640 K B of memory and a 20-40 Mb hard disk with FoxPro loaded. This configuration is purely a minimal one and is not very effective. Drawbacks could be extremely slow access and less storage. A better and preferable configuration would be an AT 386DX computer, VGA monitor, greater than 120 Mb hard disk, 640 KB base memory, above 2 Mb extended random access memory and a Fox Pro compiler. A printer is purely optional, though highly recommended for on-line library operations.

3.2 Main Modules used

- Addition Module
- Deletion Module
- Sorting Module
- Help Module
- Book-Return Module
- Lending Module

3.3 Description of Modules

1. Addition Module:

This module is used to enter new books or new members. The user should enter the information of the book for addition of a book. For entering new members, member details should be furnished. This module mainly accesses the member database for addition of members and book database for addition of books.

2. Deletion Module:

The deletion module uses a single particular key for deletion of member or a book. In case of a book the key used is the accession number and for member deletion the Roll Number is taken as the key. Using a single key facilitates both access and maintenance. This module primarily accesses the member database for deletion of members and book database for deletion of books.

3. Sorting Module:

The sorting module involves the sorting of books based on any particular field. For members, it could be on the name or Roll number. For sorting books it could be on the name, author, subject, accession number.. etc. A temporary file is used, to which the sorted new sorted file is written on to. The sorting module accesses member databases for sorting based on member fields. For sorting based on book fields it accesses the book databases.

5. Lending Module:

The lending module maintains the circulation database of the systems. People or members returning books get their names entered into the lending-circulation database. The name, roll number, the account of books, the fine due..etc. get updated automatically. Fine receipts are issued once a fine has been paid. A fine can be paid immediately or could be summed up and paid towards the end. The lending module accesses all the three databases for cross checking of information. This module and the borrowing module are the longest and most complicated of all the modules. Cross checking of information for verification of validity of both the user details and the book details makes it more difficult to assimilate.

6. Book-Return Module:

The borrowing module also helps in the maintenance of the circulation database. Once books are borrowed by members of the library, their names are entered into their roll cards and an account is kept of the book issued, roll no, number of books and date issued. This module is similar to the lending module. The only difference is that this aids in the return of the book by a member. The process of access of databases is almost the same for both. The book return-module controls the entire database and is very important for full and perfect ordering of the books. Updation and deletion are periodically carried out to ensure perfect ordering of the books.

7. Help Module.

The help module provides on-line help for each of the topics. Detailed help explaining the steps for each process have been illustrated with the aid of pop-up menus and windows. On-line help facilitates even the most tyro user who has no idea of the software.

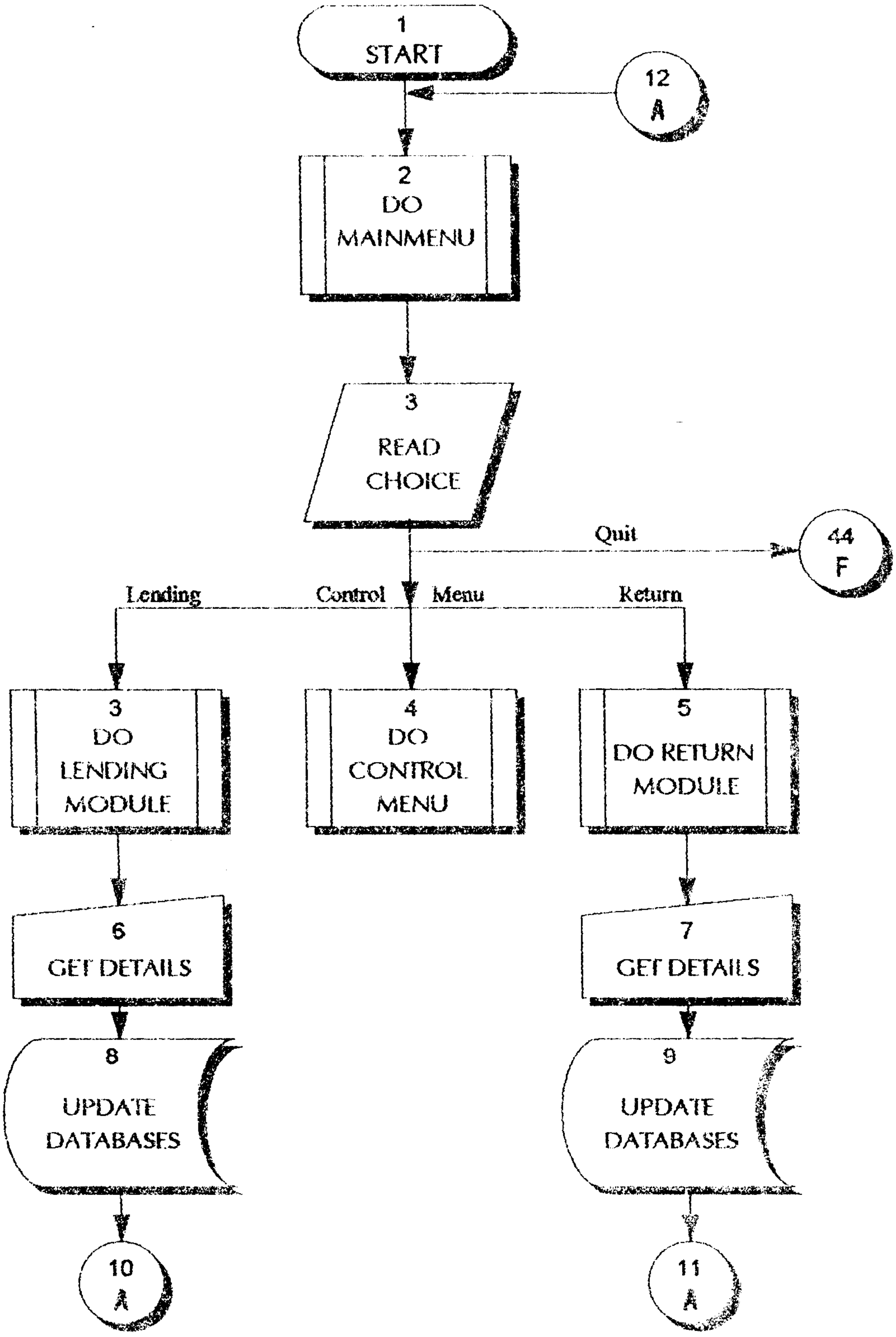

1
START
12
A
2
DO
MAINMENU
3
READ
CHOICE
Quit
44
F
Lending
Control
Menu
Return
3
DO
LENDING
MODULE
4
DO
CONTROL
MENU
5
DO RETURN
MODULE
6
GET DETAILS
7
GET DETAILS
8
UPDATE
DATABASES
9
UPDATE
DATABASES
10
A
11
A

Chart 1

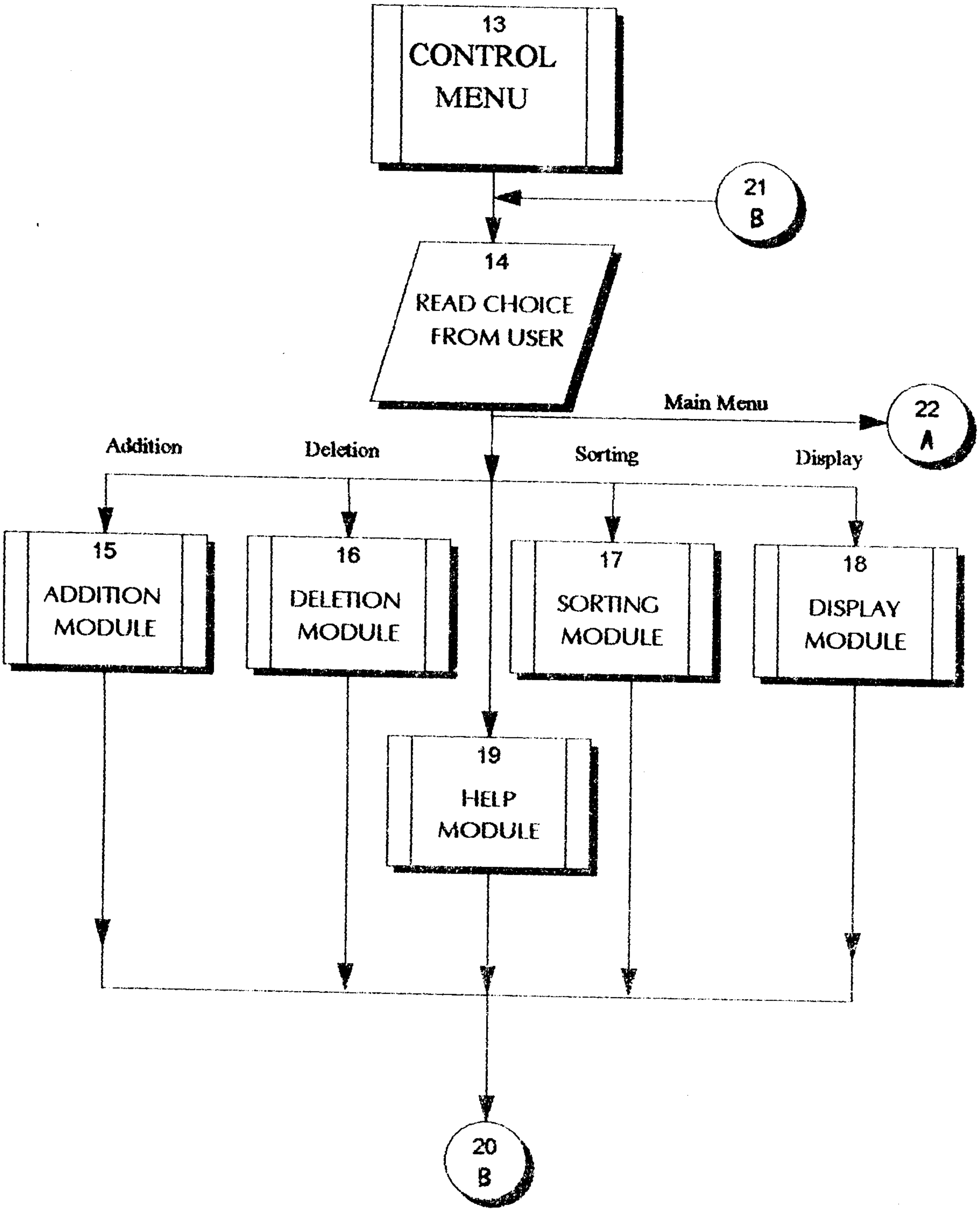

13
CONTROL MENU
21 B
14
READ CHOICE FROM USER
Main Menu
22 A
Addition
Deletion
Sorting
Display
15
ADDITION MODULE
16
DELETION MODULE
17
SORTING MODULE
18
DISPLAY MODULE
19
HELP MODULE
20 B

Chart 2

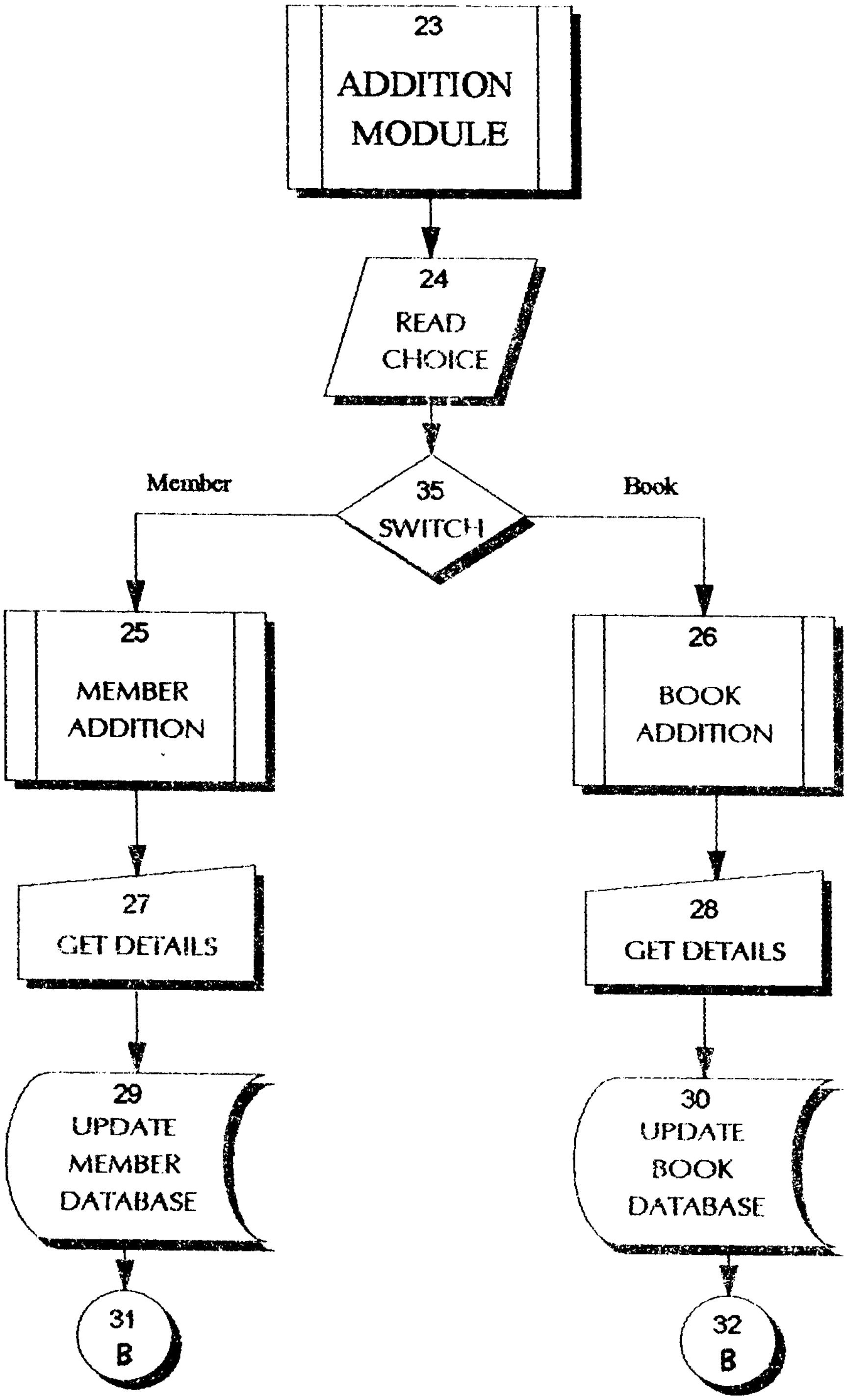

23
ADDITION MODULE
24
READ CHOICE
35
SWITCH
Member
Book
25
MEMBER ADDITION
26
BOOK ADDITION
27
GET DETAILS
28
GET DETAILS
29
UPDATE MEMBER DATABASE
30
UPDATE BOOK DATABASE
31
B
32
B

Chart 3

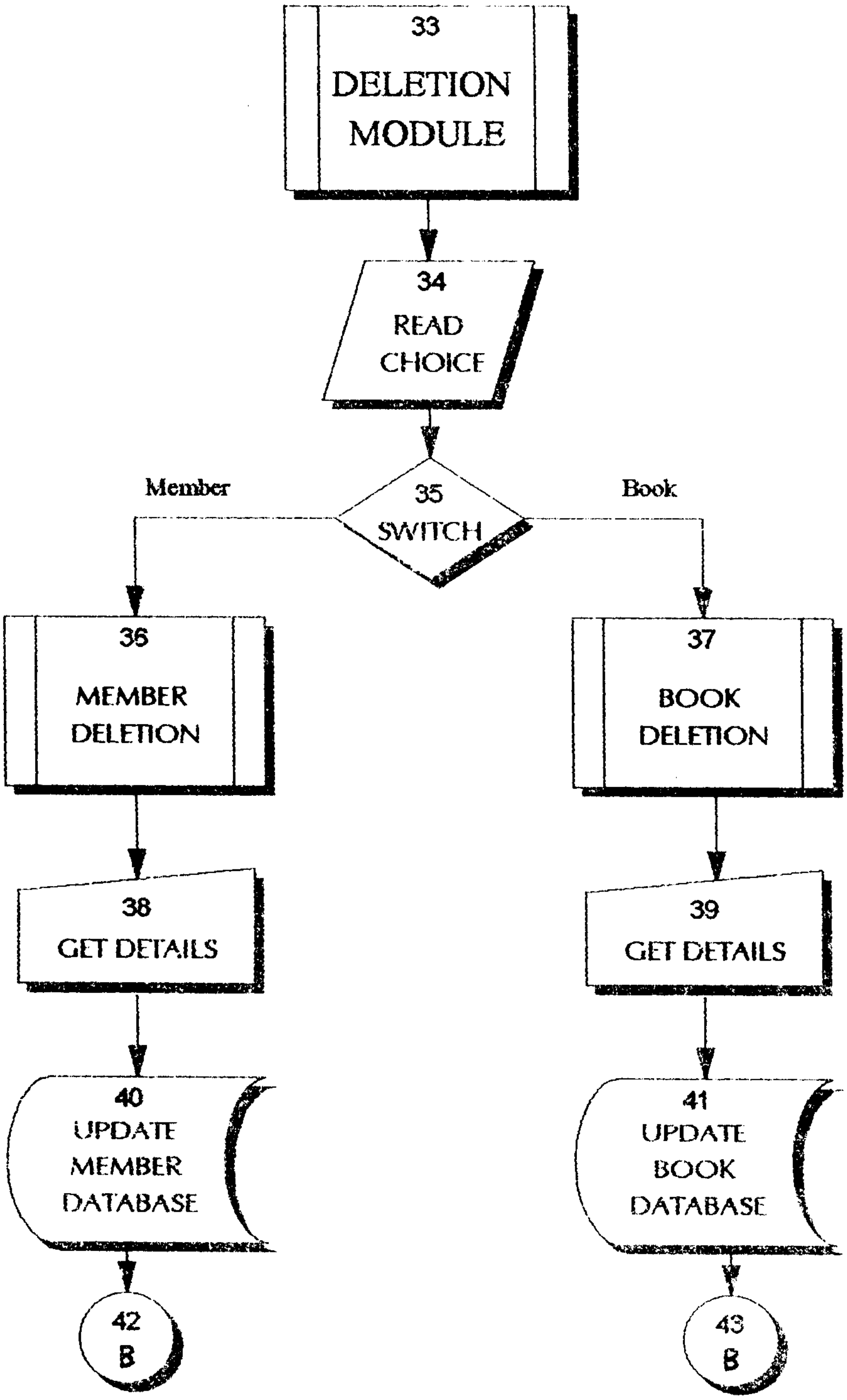

33
DELETION MODULE
34
READ CHOICE
35
SWITCH
Member
Book
36
MEMBER DELETION
37
BOOK DELETION
38
GET DETAILS
39
GET DETAILS
40
UPDATE MEMBER DATABASE
41
UPDATE BOOK DATABASE
42
B
43
B

Chart 4

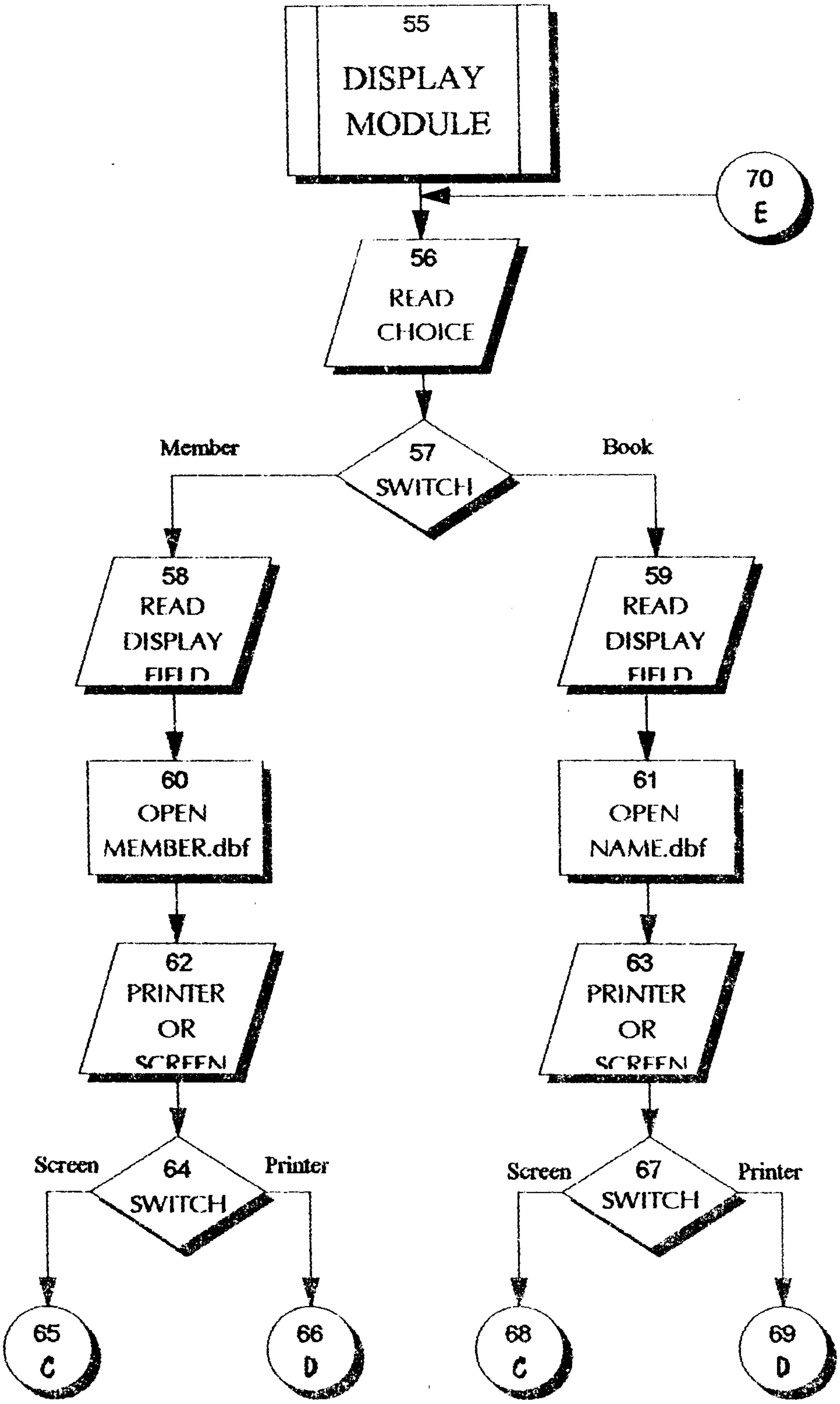

55
DISPLAY MODULE
70
E
56
READ CHOICE
Member
57
SWITCH
Book
58
READ DISPLAY FIELD
59
READ DISPLAY FIELD
60
OPEN MEMBER.dbf
61
OPEN NAME.dbf
62
PRINTER OR SCREEN
63
PRINTER OR SCREEN
Screen
64
SWITCH
Printer
Screen
67
SWITCH
Printer
65
C
66
D
68
C
69
D

Chart 6

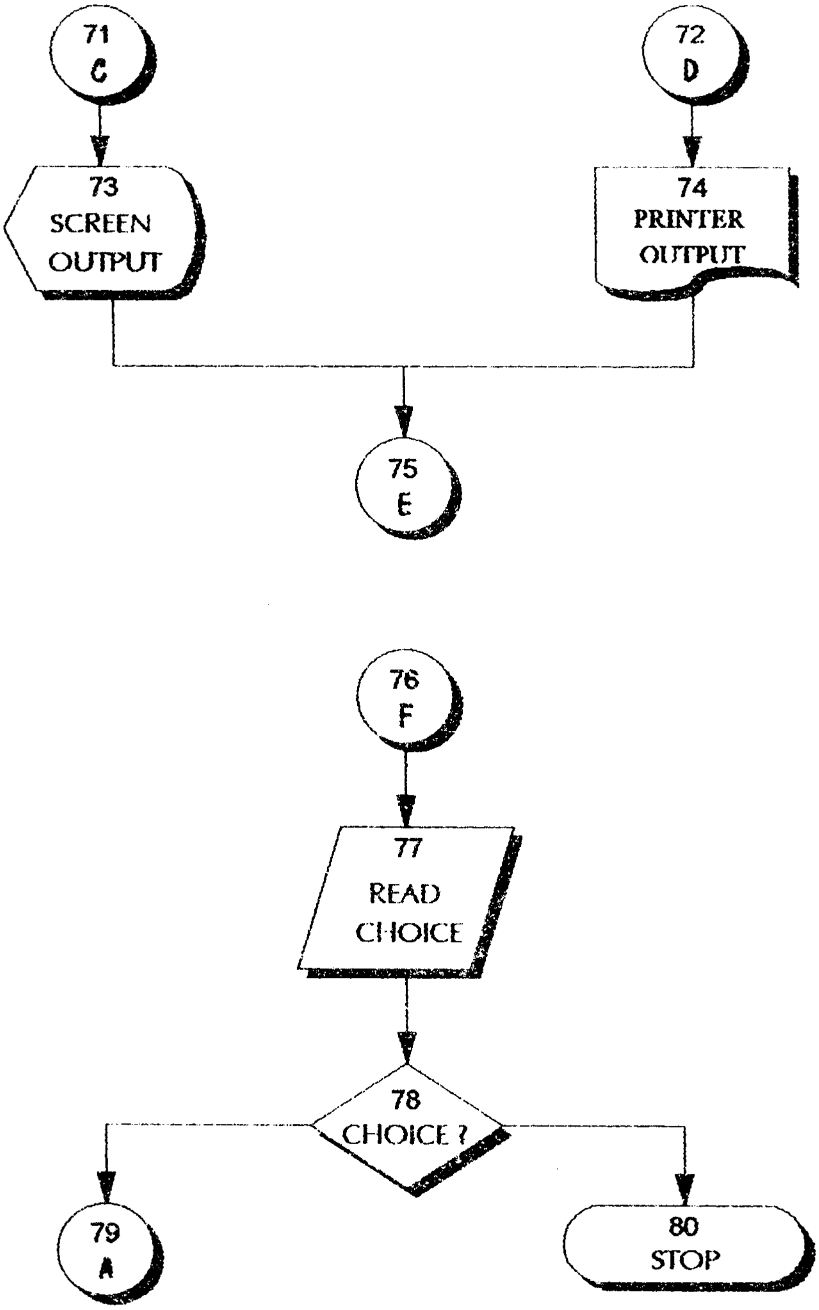

71
C
72
D
73
SCREEN
OUTPUT
74
PRINTER
OUTPUT
75
E
76
F
77
READ
CHOICE
78
CHOICE ?
79
A
80
STOP

Chart 7

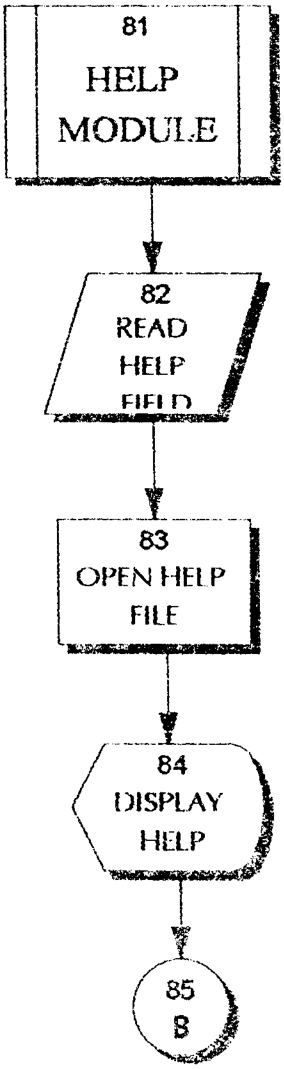

81
HELP MODULE
82
READ HELP FIELD
83
OPEN HELP FILE
84
DISPLAY HELP
85
8

Chart 8

CONCLUSION

We were able to achieve the aim of this case study,. which was to create a library management system with all major facilities. The case study was mainly custom designed for our college library. This very reason instigated into us a feeling of inspiration and motivation that was very important for our success. We fully feel that this system would be highly efficient and extremely practical for our library.

Another very simple theory that we could prove was that utilization of computers in library management produced results that had no match with the human work. Jobs like sorting and classification was done in a matter of seconds. These same jobs took hours when done by humans.

The only drawback we faced was speed. As the database grew in size, the access time and secondary storage posed a great problem. The only way out is to have a higher configuration system.

FUTURE SCOPE

SCOPE FOR FURTHER EXPANSION

We feel that the case study could have been even better if we had built it under a graphics environment like windows.

Under a Graphic User Interface a higher level of interactivity was possible with icons which picturized the entire process of understanding. The entire process of appending the database could be beautified. For example, we could store the member photos and their details into the member database. This ensures proper identification. Book covers could be scanned using a scanner and stored in the book database. One need not go into the library to search for a book.

Such a system would be a librarians paradise. Going even further we could have the job of the librarian being done by the computer. This would involve speech and character recognition, an area still under research. But, all this requires a lot of money and a higher configuration system and moreover, it is not practical in a country like ours.

BIBLIOGRAPHY

1. Illustrated Fox Pro by Robert Granillo, BPB publications.

2. Fox Pro in a day by Rayman.

3. The basics of Library Science by Mary Stivens, McGraw Hill.

SOFTWARE USED

1. Microsoft Fox Pro -- Ver 2.0 for LAN.

2. Microsoft Word -- Ver 6.0 for Windows.

3. Micrografx ABC Flow Charter -- Ver 2.0 for Windows.

4. Instant Artist

```
System: LIBRARY MANAGEMENT SYSTEM
Author: HARIHARAN. S, JITESH MENON. P, KANNAN. N
10/20/94   15:27:10
Database Structure Summary

--------------------------------------------------------------------

4 databases in the system
   MEMBER.DBF
   NAME.DBF
   CIRC.DBF
   TEMP.DBF (Temporary swap database)

--------------------------------------------------------------------

Structure for database : MEMBER.DBF
Number of data records :        13
          Last updated : 10/07/94 at 13:01
Field  Field name  Type          Width     Dec      Start       End
   1  NAME        Character        25                   1        25
   2  CARD_NO     Numeric           5                  26        30
   3  ROLL_NO     Character         7                  31        37
   4  F_OR_S      Character         1                  38        38
   5  MAX_BKS     Numeric           1                  39        39
   6  AC_BNK      Character         1                  40        40
* Total **                         41

Used by: BORBOOK.PRG
       : RETBOOK.PRG
       : CASE_SM2                          (procedure in DEFMENU.PRG)
       : ADDMEM.PRG
       : LISTMEMN.PRG
       : LISTMEMR.PRG
       : LISTMEMC.PRG
       : DELMEMB.PRG
       : DELMEMR.PRG

--------------------------------------------------------------------

Structure for database : NAME.DBF
Number of data records :        25
          Last updated : 10/07/94 at 15:28
Field  Field name  Type          Width     Dec      Start       End
    1  BOOK_NAME   Character        75                   1        75
    2  AUTHOR      Character        30                  76       105
    3  VOLUME      Numeric           3                 106       108
    4  EDITION     Character        10                 109       118
    5  SUBJECT     Character        15                 119       133
    6  COPIES      Numeric           2                 134       135
    7  T_OR_REF    Character         1                 136       136
    8  BNK_OR_GNR  Character         1                 137       137
    9  CALL_NO     Character        10                 138       147
   10  ACCN_NO     Numeric          10                 148       157
   11  STATUS      Character         1                 158       158
   12  PUBLISHER   Character        30                 159       188
   13  PRICE       Numeric           7       2         189       195
   14  DATE        Date              8                 196       203
   15  AGENTS      Character        30                 204       233
   16  BILL_NO     Numeric           6                 234       239
** Total **                        240

Used by: BORBOOK.PRG
```

```
        : CASE_SM1                   (procedure in DEFMENU.PRG)
        : ADDBOOK.PRG
        : DELBOOK.PRG
        : SORTBOOK.PRG
        : SORTMEM.PRG
        : SERBOOKN.PRG
        : SERBOOKA.PRG
```

```
Structure for database : CIRC.DBF
Number of data records :        5
     Last updated : 10/07/94 at 12:57
Field  Field name   Type        Width    Dec     Start       End
    1  ROLL_NO      Character       7                1         7
    2  TOT_BOOKS    Numeric         1                8         8
    3  ACCN_NO_1    Numeric        10                9        18
    4  ACCN_NO_2    Numeric        10               19        28
    5  ACCN_NO_3    Numeric        10               29        38
    6  ACCN_NO_4    Numeric        10               39        48
    7  ACCN_NO_5    Numeric        10               49        58
    8  DUE_DATE_1   Character       8               59        66
    9  DUE_DATE_2   Character       8               67        74
   10  DUE_DATE_3   Character       8               75        82
   11  DUE_DATE_4   Character       8               83        90
   12  DUE_DATE_5   Character       8               91        98
   13  FINE         Numeric         5      2        99       103
** Total **                       104

Used by: BORBOOK.PRG
       : RETBOOK.PRG
```

```
Structure for database : TEMP.DBF
Number of data records :       17
     Last updated : 10/07/94 at 13:16
Field  Field name   Type        Width    Dec     Start       End
    1  BOOK_NAME    Character      75                1        75
    2  AUTHOR       Character      30               76       105
    3  VOLUME       Numeric         3              106       108
    4  EDITION      Character      10              109       118
    5  SUBJECT      Character      15              119       133
    6  COPIES       Numeric         2              134       135
    7  T_OR_REF     Character       1              136       136
    8  BNK_OR_GNR   Character       1              137       137
    9  CALL_NO      Character      10              138       147
   10  ACCN_NO      Numeric        10              148       157
   11  STATUS       Character       1              158       158
   12  PUBLISHER    Character      30              159       188
   13  PRICE        Numeric         7      2       189       195
   14  DATE         Date            8              196       203
   15  AGENTS       Character      30              204       233
   16  BILL_NO      Numeric         6              234       239
** Total **                       240
```

```
System: LIBRARY MANAGEMENT SYSTEM
Author: HARIHARAN. S, JITESH MENON. P, KANNAN. N
10/20/94   15:27:10
Tree Diagram

--------------------------------------------------------------


MAINMENU.PRG
├────BORBOOK.PRG
│    │    MEMBER.DBF (database)
│    │    NAME.DBF (database)
│    │    CIRC.DBF (database)
│    ├────CAPSLOCK() (function  in FoxPro)
│    ├────UPPER() (function  in FoxPro)
│    ├────EOF() (function  in FoxPro)
│    ├────DTOC() (function  in FoxPro)
│    └────DATE() (function  in FoxPro)
├────RETBOOK.PRG
│    │    MEMBER.DBF (database)
│    │    NAME.DBF (database)
│    │    CIRC.DBF (database)
│    ├────CAPSLOCK() ... (function  in FoxPro)
│    ├────UPPER() ... (function  in FoxPro)
│    ├────EOF() ... (function  in FoxPro)
│    ├────DATE() ... (function  in FoxPro)
│    ├────CTOD() (function  in FoxPro)
│    └────CHR() (function  in FoxPro)
└────DEFMENU.PRG
     ├────MENU() (function  in FoxPro)
     ├────PAD() (function  in FoxPro)
     ├────PROMPT() (function  in FoxPro)
     ├────BOOKPROC      (procedure in DEFMENU.PRG)
     │    ├────ADDBOOK.PRG
     │    │    │    NAME.DBF (database)
     │    │    ├────CAPSLOCK() ... (function  in FoxPro)
     │    │    ├────REPLICATE() (function  in FoxPro)
     │    │    └────CHR() ... (function  in FoxPro)
     │    ├────DELBOOK.PRG
     │    │    │    NAME.DBF (database)
     │    │    └────FOUND() (function  in FoxPro)
     │    ├────SUB_FOR_SER     (procedure in DEFMENU.PRG)
     │    │    ├────PROMPT() ... (function  in FoxPro)
     │    │    └────CASE_SM1     (procedure in DEFMENU.PRG)
     │    │         │    NAME.DBF (database)
     │    │         │    TEMP.DBF (database)
     │    │         ├────SERBOOKN.PRG
     │    │         │    │    NAME.DBF (database)
     │    │         │    ├────UPPER() ... (function  in FoxPro)
     │    │         │    ├────EOF() ... (function  in FoxPro)
     │    │         │    └────CHR() ... (function  in FoxPro)
     │    │         └────SERBOOKA.PRG
     │    │              │    NAME.DBF (database)
     │    │              ├────UPPER() ... (function  in FoxPro)
     │    │              ├────EOF() ... (function  in FoxPro)
     │    │              └────CHR() ... (function  in FoxPro)
     │    └────SORTBOOK.PRG
     │         │    NAME.DBF (database)
     │         ├────PROMPT() ... (function  in FoxPro)
     │         └────CASE_SM1...     (procedure in DEFMENU.PRG)
     ├────MEMPROC      (procedure in DEFMENU.PRG)
     │    ├────ADDMEM.PRG
│         │         │         MEMBER.DBF (database)
│         │         ├─────────CAPSLOCK() ... (function  in FoxPro)
│         │         └─────────CHR() ... (function  in FoxPro)
│         ├─────────SUB_FOR_DEL     (procedure in DEFMENU.PRG)
│         │         ├─────────PROMPT() ... (function  in FoxPro)
│         │         └─────────CASE_SM3     (procedure in DEFMENU.PRG)
│         │                   ├─────────DELMEMB.PRG
│         │                   │         │         MEMBER.DBF (database)
│         │                   │         ├─────────UPPER() ... (function  in FoxPro)
│         │                   │         └─────────FOUND() ... (function  in FoxPro)
│         │                   └─────────DELMEMR.PRG
│         │                             │         MEMBER.DBF (database)
│         │                             └─────────FOUND() ... (function  in FoxPro)
│         ├─────────SUB_FOR_LIST     (procedure in DEFMENU.PRG)
│         │         ├─────────PROMPT() ... (function  in FoxPro)
│         │         └─────────CASE_SM2     (procedure in DEFMENU.PRG)
│         │                   │         MEMBER.DBF (database)
│         │                   │         TEMP1.DBF (database)
│         │                   │         TEM1P1.DBF (database)
│         │                   ├─────────LISTMEMN.PRG
│         │                   │         │         MEMBER.DBF (database)
│         │                   │         ├─────────UPPER() ... (function  in FoxPro)
│         │                   │         ├─────────EOF() ... (function  in FoxPro)
│         │                   │         └─────────CHR() ... (function  in FoxPro)
│         │                   ├─────────LISTMEMR.PRG
│         │                   │         │         MEMBER.DBF (database)
│         │                   │         ├─────────UPPER() ... (function  in FoxPro)
│         │                   │         ├─────────EOF() ... (function  in FoxPro)
│         │                   │         └─────────CHR() ... (function  in FoxPro)
│         │                   └─────────LISTMEMC.PRG
│         │                             │         MEMBER.DBF (database)
│         │                             ├─────────EOF() ... (function  in FoxPro)
│         │                             └─────────CHR() ... (function  in FoxPro)
│         └─────────SORTMEM.PRG
│                   │         NAME.DBF (database)
│                   ├─────────PROMPT() ... (function  in FoxPro)
│                   └─────────CASE_SM1...     (procedure in DEFMENU.PRG)
├─────────HELPPROC     (procedure in DEFMENU.PRG)
└─────────STOPROC     (procedure in DEFMENU.PRG)
```

```
*:*********************************************************************
*****************
*:
*:          Program: BORBOOK.PRG
*:
*:           System: LIBRARY MANAGEMENT SYSTEM
*:           Author: HARIHARAN. S, JITESH MENON. P, KANNAN.
N
*:        Copyright (c) 1994, ANONYMOUS
*:    Last modified: 10/20/94     14:30
*:
*:        Called by: MAINMENU.PRG
*:
*:            Calls: CAPSLOCK()          (function  in ?)
*:                 : UPPER()             (function  in ?)
*:                 : EOF()               (function  in ?)
*:                 : DTOC()              (function  in ?)
*:                 : DATE()              (function  in ?)
*:
*:             Uses: MEMBER.DBF
*:                 : NAME.DBF
*:                 : CIRC.DBF
*:
*:        Documented 10/20/94 at 14:45                FoxDoc
 version 2.10e
*:*********************************************************************
****************

*BORBOOK.PRG
*Module for borrowing a book

SET TALK OFF
CAPSTATUS = CAPSLOCK(.T.)
DEFINE WINDOW SERWIN1 FROM 03,05 TO 15, 60 SHADOW TITLE "B
orrow a book"
ACTIVATE WINDOW SERWIN1
*DO WHILE .T.
*----------------------------------------------------------------------
--------

*Searches for the Roll Number in the member database
CLEAR
@ 0,0 SAY 'Enter the Roll No. of the student or press Q to
 quit'
ACCEPT '> 'TO M.ROL
CLEAR
```

```
     *Searches for the Roll Number in the member database
     CLEAR
     @ 0,0 SAY 'Enter the Roll No. of the student or press Q to quit'
     ACCEPT '> 'TO M.ROL
     CLEAR
  ┌─IF UPPER(M.ROL) = 'Q'
  │      CLEAR
  │      DEACTIVATE WINDOW SERWIN1
<────────RETURN
  └─ENDIF

     USE MEMBER

     LOCATE FOR UPPER(ROLL_NO) = UPPER(M.ROL)
  ┌─IF EOF()
  │      ? "Cannot find student roll number ", M.ROL
  │      WAIT
  │      DEACTIVATE WINDOW SERWIN1
<────────RETURN
  └─ENDIF
     STORE MAX_BKS TO MAX
     *----------------------------------------------------------------------
     *----------------------------------------------------------------------
     *Searches for Accn. No in the book database

     @ 1,0 SAY 'Enter the Accn. No. of the book or press 0 to quit'
     INPUT '> 'TO MNO
     CLEAR
  ┌─IF MNO = 0
  │      CLEAR
  │      DEACTIVATE WINDOW SERWIN1
<────────RETURN
  └─ENDIF

     USE NAME

     LOCATE FOR ACCN_NO = MNO
  ┌─IF EOF()
  │      ? "Cannot find given Accn. No. of the book", MNO
  │      WAIT
  │      DEACTIVATE WINDOW SERWIN1
<────────RETURN
  └─ENDIF

  ┌─IF UPPER(STATUS) = 'N'
  │      ? "Book has already been borrowed."
  │      WAIT
  │      DEACTIVATE WINDOW SERWIN1
<────────RETURN
  └─ENDIF
     REPLACE STATUS WITH 'N'
     *----------------------------------------------------------------------
     *----------------------------------------------------------------------
     *Searches the circulation database to check whether there is an
     *existing record for this roll number

     USE CIRC

     LOCATE FOR UPPER(ROLL_NO) = UPPER(M.ROL)
```

```
 ┌─IF EOF()
 │    GO BOTTOM
 │    APPEND BLANK
 └─ENDIF
   *-----------------------------------------------------------------------------------------------------------
   *-----------------------------------------------------------------------------------------------------------

  REPLACE ROLL_NO WITH M.ROL
┌─IF TOT_BOOKS < MAX
│  ┌─DO CASE
│  ├─CASE ACCN_NO_1 = 0
│  │     REPLACE DUE_DATE_1 WITH DTOC(DATE() + 15)
│  │     REPLACE ACCN_NO_1 WITH MNO
│  ├─CASE ACCN_NO_2 = 0
│  │     REPLACE DUE_DATE_2 WITH DTOC(DATE() + 15)
│  │     REPLACE ACCN_NO_2 WITH MNO
│  ├─CASE ACCN_NO_3 = 0
│  │     REPLACE DUE_DATE_3 WITH DTOC(DATE() + 15)
│  │     REPLACE ACCN_NO_3 WITH MNO
│  ├─CASE ACCN_NO_4 = 0
│  │     REPLACE DUE_DATE_4 WITH DTOC(DATE() + 15)
│  │     REPLACE ACCN_NO_4 WITH MNO
│  ├─CASE ACCN_NO_5 = 0
│  │     REPLACE DUE_DATE_5 WITH DTOC(DATE() + 15)
│  │     REPLACE ACCN_NO_5 WITH MNO
│  └─ENDCASE
│     TOT = TOT_BOOKS + 1
│     REPLACE TOT_BOOKS WITH TOT
├─ELSE
│     CLEAR
│     @ 0,1 SAY 'Roll number '+M.ROL+' has reached the maximum limit'
│     @ 1,1 SAY 'His/Her maximum limit is '
│     @ 1,26 SAY MAX
│     WAIT
└─ENDIF
  DEACTIVATE WINDOW SERWIN1
  CAPSTATUS = CAPSLOCK(.T.)
  SET TALK ON
  SET HEADING ON
  CLOSE DATABASES
  CLEAR
  RETURN

 *: EOF: BORBOOK.PRG
```

```
*:******************************************************************************
*:
*:         Program:                 DELMEMB.PRG
*:
*:          System: LIBRARY MANAGEMENT SYSTEM
*:          Author: HARIHARAN. S, JITESH MENON. P, KANNAN. N
*:       Copyright (c) 1994, ANONYMOUS
*:  Last modified: 10/20/94     14:58
*:
*:       Called by: CASE_SM3          (procedure in DEFMENU.PRG)
*:
*:           Calls: UPPER()           (function  in ?)
*:                : FOUND()           (function  in ?)
*:
*:            Uses: MEMBER.DBF
*:
*:       Documented 10/20/94 at 15:27                FoxDoc  version 2.10e
*:******************************************************************************

     SET TALK OFF
     USE MEMBER
     DEFINE WINDOW WIN1 FROM 03,05 TO 22,70 SHADOW TITLE "Deletion of a member"
     ACTIVATE WINDOW WIN1
     CLEAR
     INPUT 'How many members do you want to delete : ' TO MEMNO
     CLEAR
   ┌─FOR I=1 TO MEMNO
   │    CLEAR
   │    ? 'Member',I
   │    ? '------'
   │    ?
   │    INPUT "Enter the member name (exactly as entered) : " TO M.MNAME
   │    ACCEPT "Are you sure (Y/N) ?  " TO M.CHOICE
   │    ?
   │  ┌─IF LOWER (M.CHOICE) = 'y'
   │  │    LOCATE FOR UPPER(NAME) = UPPER(M.MNAME)
   │  │  ┌─IF FOUND() = .F.
   │  │  │    ? "Cannot find member with given name", M.ROLL
   │  │  │    ? "Record Not deleted, Try again !!"
   │  │  │    WAIT
   │  │  │    ?
   v──────────EXIT
   │  │  ├─ELSE
   │  │  │    DELETE FOR UPPER(NAME) = UPPER(M.MNAME)
   │  │  │    ? "Specified member record deleted."
   │  │  │    ?
   │  │  │    WAIT
   │  │  └─ENDIF
   │  └─ENDIF
   │  ┌─IF LOWER (M.CHOICE) = 'n'
   │  │    ? "Record has not been deleted"
   │  │    ?
   │  │    WAIT
   │  └─ENDIF
   └─ENDFOR
     PACK
     DEACTIVATE WINDOW WIN1
```

```
      CLOSE DATABASES
      RETURN

      *: EOF: DELMEMR.ACT
```

```
 *:**********************************************************************************
******
 *:
 *:
 *:           Program: DELMEMR.PRG
 *:
 *:            System: LIBRARY MANAGEMENT SYSTEM
 *:            Author: HARIHARAN. S, JITESH MENON. P, KANNAN. N
 *:         Copyright (c) 1994, ANONYMOUS
 *:     Last modified: 10/20/94     14:58
 *:
 *:         Called by: CASE_SM3                    (procedure in DEFMENU.PRG)
 *:
 *:             Calls: FOUND()                     (function  in ?)
 *:
 *:              Uses: MEMBER.DBF
 *:
 *:        Documented 10/20/94 at 15:27                        FoxDoc  version 2.10
 *:**********************************************************************************
******

      SET TALK OFF
      USE MEMBER
      DEFINE WINDOW WIN1 FROM 03,05 TO 22,70 SHADOW TITLE "Deletion of a m
mber"
      ACTIVATE WINDOW WIN1
      CLEAR
      INPUT 'How many members do you want to delete : ' TO MEMNO
      CLEAR
    ┌─FOR I=1 TO MEMNO
    │    CLEAR
    │    ? 'Member',I
    │    ? '———————'
    │    ?
    │    INPUT "Enter the Roll Number for the book : " TO M.ROLL
    │    ACCEPT "Are you sure (Y/N) ?  " TO M.CHOICE
    │    ?
    │ ┌─IF LOWER (M.CHOICE) = 'y'
    │ │    LOCATE FOR ROLL_NO = M.ROLL
    │ │ ┌─IF FOUND() = .F.
    │ │ │    ? "Cannot find member with given Roll number ", M.ROLL
    │ │ │    ? "Record Not deleted, Try again !!"
    │ │ │    WAIT
    │ │ │    ?
    v────────EXIT
    │ │ ├─ELSE
    │ │ │    DELETE FOR ROLL_NO = M.ROLL
    │ │ │    ? "Specified member record deleted."
    │ │ │    ?
```

```
        ON SELECTION POPUP POPQUIT DO STOPROC

        *Display and activate the menu bar
        ACTIVATE MENU MAINMENU
        SET TALK ON



        *------------------------------------------------------------------------
   *!************************************************************************
******
   *!
   *!      Procedure: BOOKPROC
   *!
   *!      Called by: DEFMENU.PRG
   *!
   *!          Calls: ADDBOOK.PRG
   *!               : DELBOOK.PRG
   *!               : SUB_FOR_SER           (procedure in DEFMENU.PRG)
   *!               : SORTBOOK.PRG
   *!
   *!************************************************************************
******
        PROCEDURE BOOKPROC
        *Display the names of the pad and the menu
        PARAMETER MMENU, MPAD, MPROMPT

     ┌──DO CASE
     ├──CASE MPROMPT = 'Add a book'
     │     DO ADDBOOK
     ├──CASE MPROMPT = 'Delete a book'
     │     DO DELBOOK
     ├──CASE MPROMPT = 'Search for a book'
     │     DO SUB_FOR_SER
     ├──CASE MPROMPT = 'Sort books'
     │     DO SORTBOOK
     └──ENDCASE
        RETURN
        *------------------------------------------------------------------------

        *------------------------------------------------------------------------
   *!************************************************************************
******
   *!
   *!      Procedure: MEMPROC
   *!
   *!      Called by: DEFMENU.PRG
   *!
   *!          Calls: ADDMEM.PRG
   *!               : SUB_FOR_DEL           (procedure in DEFMENU.PRG)
   *!               : SUB_FOR_LIST          (procedure in DEFMENU.PRG)
   *!               : SORTMEM.PRG
   *!
   *!************************************************************************
******
        PROCEDURE MEMPROC
        *Display the names of the pad and the menu
        PARAMETER MMENU, MPAD, MPROMPT

     ┌──DO CASE
     ├──CASE MPROMPT = 'Add a member'
     │     DO ADDMEM
     ├──CASE MPROMPT = 'Delete a member'
     │     DO SUB_FOR_DEL
```

```
      ├─CASE MPROMPT = 'List members'
      │    DO SUB_FOR_LIST
      ├─CASE MPROMPT = 'Sort members'
      │    DO SORTMEM
      └─ENDCASE
        RETURN
        *-----------------------------------------------------------------

        *-----------------------------------------------------------------
 *!*************************************************************************
******
 *!
 *!      Procedure: HELPPROC
 *!
 *!      Called by: DEFMENU.PRG
 *!
 *!*************************************************************************
******
        PROCEDURE HELPPROC
        *Display the names of the pad and the menu
        PARAMETER MMENU, MPAD, MPROMPT

      ┌─DO CASE
      ├─CASE MPROMPT = 'Lending'
      │    DEFINE WINDOW HELP FROM 10,00 TO 15, 60 CLOSE
      │    SHADOW COLOR SCHEME 1 TITLE "Help - Lending" DOUBLE
      │    ACTIVATE WINDOW HELP
      │    @ 0,0 SAY 'Use this command for lending a book to a member.'
      │    @ 1,0 SAY 'The inputs it requires are the the Roll No. and Accn.
o'
      │    WAIT
      │    DEACTIVATE WINDOW HELP
      ├─CASE MPROMPT = 'Receiving'
      │    DEFINE WINDOW HELP FROM 10,00 TO 15, 60 CLOSE
      │    SHADOW COLOR SCHEME 1 TITLE "Help - Receiving" DOUBLE
      │    ACTIVATE WINDOW HELP
      │    @ 0,0 SAY 'Use this command for receiving a book from a member.'
      │    @ 1,0 SAY 'The inputs it requires are the the Roll No. and Accn.
o'
      │    WAIT
      │    DEACTIVATE WINDOW HELP
      ├─CASE MPROMPT = 'Members'
      │    DEFINE WINDOW HELP FROM 10,00 TO 15, 60 CLOSE
      │    SHADOW COLOR SCHEME 1 TITLE "Help - Members" DOUBLE
      │    ACTIVATE WINDOW HELP
      │    @ 0,0 SAY 'Members are the people who have rights to borrow'
      │    @ 1,0 SAY 'or use the library and its facilities'
      │    WAIT
      │    DEACTIVATE WINDOW HELP
      ├─CASE MPROMPT = 'Books'
      │    DEFINE WINDOW HELP FROM 10,00 TO 15, 60 CLOSE
      │    SHADOW COLOR SCHEME 1 TITLE "Help - Books" DOUBLE
      │    ACTIVATE WINDOW HELP
      │    @ 0,0 SAY 'Books are the entities of any Book Library'
      │    @ 1,0 SAY 'These entities can either be borrowed or returned.'
      │    WAIT
      │    DEACTIVATE WINDOW HELP
      ├─CASE MPROMPT = 'Reservation'
      │    DEFINE WINDOW HELP FROM 10,00 TO 15, 60 CLOSE
```

```
    │     SHADOW COLOR SCHEME 1 TITLE "Help - Reservation" DOUBLE
    │     ACTIVATE WINDOW HELP
    │     @ 0,0 SAY 'Use this command for reserving a book.'
    │     @ 1,0 SAY 'A member may reserve a book for future borrowing.'
    │     WAIT
    │     DEACTIVATE WINDOW HELP
    ├─CASE MPROMPT = 'Sorting'
    │     DEFINE WINDOW HELP FROM 10,00 TO 15, 60 CLOSE
    │     SHADOW COLOR SCHEME 1 TITLE "Help - Sorting" DOUBLE
    │     ACTIVATE WINDOW HELP
    │     @ 0,0 SAY 'Use this command for sorting based on the given keys.'
    │     @ 1,0 SAY 'Keys could be Accn. No, Roll No., Member Name .. etc'
    │     WAIT
    │     DEACTIVATE WINDOW HELP
    ├─CASE MPROMPT = 'Addition'
    │     DEFINE WINDOW HELP FROM 10,00 TO 15, 60 CLOSE SHADOW
    │     COLOR SCHEME 1 TITLE "Help - Addition" DOUBLE
    │     ACTIVATE WINDOW HELP
    │     @ 0,0 SAY 'Use this command for Adding a member or book'
    │     @ 1,0 SAY 'The inputs it requires are simple to understand.'
    │     WAIT
    │     DEACTIVATE WINDOW HELP
    ├─CASE MPROMPT = 'Deletion'
    │     DEFINE WINDOW HELP FROM 10,00 TO 15, 60 CLOSE SHADOW
    │     COLOR SCHEME 1 TITLE "Help - Deletion" DOUBLE
    │     ACTIVATE WINDOW HELP
    │     @ 0,0 SAY 'Use this command for Deleting a book or member.'
    │     @ 1,0 SAY 'The inputs it requires are self understandable'
    │     WAIT
    │     DEACTIVATE WINDOW HELP
    ├─CASE MPROMPT = 'Searching'
    │     DEFINE WINDOW HELP FROM 10,00 TO 15, 60 CLOSE
    │     SHADOW COLOR SCHEME 1 TITLE "Help - Searching" DOUBLE
    │     ACTIVATE WINDOW HELP
    │     @ 0,0 SAY 'Use this command for Searching a book or member.'
    │     @ 1,0 SAY 'The inputs it requires are variou keys.'
    │     WAIT
    │     DEACTIVATE WINDOW HELP
    └─ENDCASE
    RETURN
    *-------------------------------------------------------------------------

    *-------------------------------------------------------------------------
*!*************************************************************************
******
*!
*!      Procedure: STOPROC
*!
*!      Called by: DEFMENU.PRG
*!
*!*************************************************************************
******
    PROCEDURE STOPROC
    *Return to the FOX pro command window
    SET CLOCK OFF
    CLEAR
    DEACTIVATE WINDOW MAIN
    DEACTIVATE POPUP POPBOOK
    DEACTIVATE POPUP POPCIR
    DEACTIVATE POPUP POPHELP
    DEACTIVATE POPUP POPQUIT
```

```
     RETURN
     *-----------------------------------------------------------------------------

     *-----------------------------------------------------------------------------
*!***************************************************************************
******
*!
*!      Procedure: SUB_FOR_SER
*!
*!      Called by: BOOKPROC              (procedure in DEFMENU.PRG)
*!
*!          Calls: PROMPT()              (function  in ?)
*!               : CASE_SM1              (procedure in DEFMENU.PRG)
*!
*!***************************************************************************
******
     PROCEDURE SUB_FOR_SER
     *Procedure for sub menu creation for searching a book
     DEFINE POPUP SUB_MENU_1 FROM 6,15 COLOR SCHEME 2 TITLE "Search" SHADO
W

     DEFINE BAR 1 OF SUB_MENU_1 PROMPT "Search using \<Book name"
     DEFINE BAR 2 OF SUB_MENU_1 PROMPT "Search using \<Author"
     ON SELECTION POPUP SUB_MENU_1 DO CASE_SM1 WITH PROMPT()
     ACTIVATE POPUP SUB_MENU_1
     RETURN


*!***************************************************************************
******
*!
*!      Procedure: CASE_SM1
*!
*!      Called by: SUB_FOR_SER           (procedure in DEFMENU.PRG)
*!               : SORTBOOK.PRG
*!               : SORTMEM.PRG
*!
*!          Calls: SERBOOKN.PRG
*!               : SERBOOKA.PRG
*!
*!           Uses: NAME.DBF
*!               : TEMP.DBF
*!
*!***************************************************************************
******
     PROCEDURE CASE_SM1
     *Procedure for case structure for submenu
     PARAMETER  MPROMPT

   ┌─DO CASE
   ├─CASE MPROMPT = 'Search using Book name'
   │    DO SERBOOKN
   ├─CASE MPROMPT = 'Search using Author'
   │    DO SERBOOKA
   │    *CASE mprompt = 'Search for a book'
   └─ENDCASE
     RETURN
     *-----------------------------------------------------------------------------
```

```
      RETURN
      *------------------------------------------------------------------------------
-

      *------------------------------------------------------------------------------
-
*!*****************************************************************************
******
*!
*!        Procedure: SUB_FOR_SER
*!
*!        Called by: BOOKPROC              (procedure in DEFMENU.PRG)
*!
*!            Calls: PROMPT()              (function  in ?)
*!                 : CASE_SM1              (procedure in DEFMENU.PRG)
*!
*!*****************************************************************************
******
      PROCEDURE SUB_FOR_SER
      *Procedure for sub menu creation for searching a book
      DEFINE POPUP SUB_MENU_1 FROM 6,15 COLOR SCHEME 2 TITLE "Search" SHADO
W

      DEFINE BAR 1 OF SUB_MENU_1 PROMPT "Search using \<Book name"
      DEFINE BAR 2 OF SUB_MENU_1 PROMPT "Search using \<Author"
      ON SELECTION POPUP SUB_MENU_1 DO CASE_SM1 WITH PROMPT()
      ACTIVATE POPUP SUB_MENU_1
      RETURN


*!*****************************************************************************
******
*!
*!        Procedure: CASE_SM1
*!
*!        Called by: SUB_FOR_SER           (procedure in DEFMENU.PRG)
*!                 : SORTBOOK.PRG
*!                 : SORTMEM.PRG
*!
*!            Calls: SERBOOKN.PRG
*!                 : SERBOOKA.PRG
*!
*!             Uses: NAME.DBF
*!                 : TEMP.DBF
*!
*!*****************************************************************************
******
      PROCEDURE CASE_SM1
      *Procedure for case structure for submenu
      PARAMETER  MPROMPT

    ┌─DO CASE
    ├─CASE MPROMPT = 'Search using Book name'
    │    DO SERBOOKN
    ├─CASE MPROMPT = 'Search using Author'
    │    DO SERBOOKA
    │    *CASE mprompt = 'Search for a book'
    └─ENDCASE
      RETURN
      *------------------------------------------------------------------------------

      *-------------------------------------------------------------------------
*!**************************************************************************
******
*!
*!      Procedure: SUB_FOR_LIST
*!
*!      Called by: MEMPROC              (procedure in DEFMENU.PRG)
*!
*!          Calls: PROMPT()             (function  in ?)
*!               : CASE_SM2             (procedure in DEFMENU.PRG)
*!
*!**************************************************************************
******
      PROCEDURE SUB_FOR_LIST
      *Procedure for sub menu creation for searching a book
      DEFINE POPUP SUB_MENU_2 FROM 6,15 COLOR SCHEME 2 TITLE "List" SHADOW

      DEFINE BAR 1 OF SUB_MENU_2 PROMPT "List by \<Name"
      DEFINE BAR 2 OF SUB_MENU_2 PROMPT "List by \<Roll-No"
      DEFINE BAR 3 OF SUB_MENU_2 PROMPT "List by \<Card-No"
      DEFINE BAR 4 OF SUB_MENU_2 PROMPT "\<Display database"
      ON SELECTION POPUP SUB_MENU_2 DO CASE_SM2 WITH PROMPT()
      ACTIVATE POPUP SUB_MENU_2
      RETURN


*!**************************************************************************
******
*!
*!      Procedure: CASE_SM2
*!
*!      Called by: SUB_FOR_LIST         (procedure in DEFMENU.PRG)
*!
*!          Calls: LISTMEMN.PRG
*!               : LISTMEMR.PRG
*!               : LISTMEMC.PRG
*!
*!           Uses: MEMBER.DBF
*!               : TEMP1.DBF
*!               : TEM1P1.DBF
*!
*!**************************************************************************
******
      PROCEDURE CASE_SM2
      *Procedure for case structure for submenu
      PARAMETER  MPROMPT

    ┌─DO CASE
    ├─CASE MPROMPT = 'List by Name'
    │    DO LISTMEMN
    ├─CASE MPROMPT = 'List by Roll-No'
    │    DO LISTMEMR
    ├─CASE MPROMPT = 'List by Card-No'
    │    DO LISTMEMC
    ├─CASE MPROMPT = 'Display database'
    │    USE MEMBER
    │    BROWSE
    │    CLOSE DATABASES
    └─ENDCASE
      RETURN
      *-------------------------------------------------------------------------

```

```
        *-----------------------------------------------------------------------

*!***********************************************************************
******
*!
*!      Procedure: SUB_FOR_DEL
*!
*!      Called by: MEMPROC               (procedure in DEFMENU.PRG)
*!
*!          Calls: PROMPT()              (function  in ?)
*!               : CASE_SM3              (procedure in DEFMENU.PRG)
*!
*!***********************************************************************
******
        PROCEDURE SUB_FOR_DEL
        *Procedure for sub menu creation for searching a book
        DEFINE POPUP SUB_MENU_3 FROM 6,20 COLOR SCHEME 2 TITLE "Deletion" SI
DOW

        DEFINE BAR 1 OF SUB_MENU_3 PROMPT "Using \<Name"
        DEFINE BAR 2 OF SUB_MENU_3 PROMPT "Using \<Roll No."
        ON SELECTION POPUP SUB_MENU_3 DO CASE_SM3 WITH PROMPT()
        ACTIVATE POPUP SUB_MENU_3
        RETURN


*!***********************************************************************
******
*!
*!      Procedure: CASE_SM3
*!
*!      Called by: SUB_FOR_DEL           (procedure in DEFMENU.PRG)
*!
*!          Calls: DELMEMB.PRG
*!               : DELMEMR.PRG
*!
*!***********************************************************************
******
        PROCEDURE CASE_SM3
        *Procedure for case structure for submenu
        PARAMETER  MPROMPT

      ┌─DO CASE
      ├─CASE MPROMPT = 'Using Name'
      │    DO DELMEMB
      ├─CASE MPROMPT = 'Using Roll No.'
      │    DO DELMEMR
      └─ENDCASE
        RETURN
        *-----------------------------------------------------------------------



        *: EOF: DEFMENU.PRG
```

```
   @ 0,23 SAY MTEMP
   @ 1,0 SAY REPLICATE(CHR(205),64)
   ?
   @ 18,0 SAY REPLICATE(CHR(205),64)
   @ 2,2 SAY 'Book Name    : ' GET BOOK_NAME
   @ 3,2 SAY 'Author       : ' GET AUTHOR
   @ 4,2 SAY 'Volume       : ' GET VOLUME
   @ 5,2 SAY 'Edition      : ' GET EDITION
   @ 6,2 SAY 'Subject      : ' GET SUBJECT
   @ 7,2 SAY 'Copies       : ' GET COPIES
   @ 8,2 SAY 'Text/Ref.    : ' GET T_OR_REF
   @ 9,2 SAY 'Bank/Gen.    : ' GET BNK_OR_GNR
   @ 10,2 SAY 'Call No.     : ' GET CALL_NO
   @ 11,2 SAY 'Accn. No.    : ' GET ACCN_NO
   @ 12,2 SAY 'Available    : ' GET STATUS
   @ 13,2 SAY 'Publisher    : ' GET PUBLISHER
   @ 14,2 SAY 'Price                @ 15,2 SAY 'Date        : ' GET

   @ 16,2 SAY 'Agents       : ' GET AGENTS
   @ 17,2 SAY 'Bill No.     : ' GET BILL_NO
   READ
ENDFOR
? "Information has been appended to Book Database"
WAIT
SET CONFIRM OFF
DEACTIVATE WINDOW WIN1
CLOSE DATABASES
RETURN

*: EOF: ADDBOOK.PRG
```

```
*:**************************************************************************
*:
*:           Program: D:\FP2\CASE\ADDBOOK.PRG
*:
*:            System: LIBRARY MANAGEMENT SYSTEM
*:            Author: HARIHARAN. S, JITESH MENON. P, KANNAN. N
*:         Copyright (c) 1994, ANONYMOUS
*:     Last modified: 10/20/94     14:52
*:
*:         Called by: BOOKPROC           (procedure in DEFMENU.PRG)
*:
*:             Calls: CAPSLOCK()         (function  in ?)
*:                  : REPLICATE()        (function  in ?)
        ?)
*:
*:              Uses: NAME.DBF
*:
*:        Documented 10/20/94 at 15:27             FoxDoc  version 2.10e
*:**************************************************************************
     *procedure for adding books

     SET TALK OFF
     SET CONFIRM ON

     USE NAME

     DEFINE WINDOW WIN1 FROM 03,05 TO 24,70 SHADOW TITLE "Addition of a book"
     ACTIVATE WINDOW WIN1
     CLEAR
     INPUT 'How many books do you want to add : ' TO BOOKNO
     CAPSTATUS = CAPSLOCK(.T.)
     CLEAR
    ┌─FOR I=1 TO BOOKNO
    │   CLEAR
    │   SET CONFIRM ON
    │   APPEND BLANK
    │   GO BOTTOM
    │   STORE I TO MTEMP
    │   @ 0,2 SAY 'Information for book '
    │   @ 0,23 SAY MTEMP
    │   @ 1,0 SAY REPLICATE(CHR(205),64)
    │   ?
    │   @ 18,0 SAY REPLICATE(CHR(205),64)
    │   @ 2,2 SAY 'Book Name   : ' GET BOOK_NAME
    │   @ 3,2 SAY 'Author      : ' GET AUTHOR
    │   @ 4,2 SAY 'Volume      : ' GET VOLUME
    │   @ 5,2 SAY 'Edition     : ' GET EDITION
    │   @ 6,2 SAY 'Subject     : ' GET SUBJECT
    │   @ 7,2 SAY 'Copies      : ' GET COPIES
    │   @ 8,2 SAY 'Text/Ref.   : ' GET T_OR_REF
       │   @ 10,2 SAY 'Call No.    : ' GET CALL_NO
    │   @ 11,2 SAY 'Accn. No.   : ' GET ACCN_NO
    │   @ 12,2 SAY 'Available   : ' GET STATUS
    │   @ 13,2 SAY 'Publisher   : ' GET PUBLISHER
    │   @ 14,2 SAY 'Price       : ' GET PRICE
    │   @ 15,2 SAY 'Date        : ' GET DATE
    │   @ 16,2 SAY 'Agents      : ' GET AGENTS
    │   @ 17,2 SAY 'Bill No.    : ' GET BILL_NO
    │   READ
    └─ENDFOR
    ? "Information has been appended to Book Databa
se"
    WAIT
    SET CONFIRM OFF
    DEACTIVATE WINDOW WIN1
    CLOSE DATABASES
    RETURN

    *: EOF: ADDBOOK.ACT
```

```
*:**********************************************************
*********************
*:
*:          Program: ADDMEM.PRG
*:
*:           System: LIBRARY MANAGEMENT SYSTEM
*:           Author: HARIHARAN. S, JITESH MENON. P, KANNAN.
N
*:        Copyright (c) 1994, ANONYMOUS
*:    Last modified: 09/17/94     12:11
*:
*:        Called by: MEMPROC              (procedure in DEFMEN
U.PRG)
*:
*:            Calls: CAPSLOCK()           (function  in ?)
*:                 : CHR()                (function  in ?)
*:
*:             Uses: MEMBER.DBF
*:
*:       Documented 10/20/94 at 14:45                        version 2
*:**********************************************************
*********************
*procedure for adding members

SET TALK OFF
CAPSTATUS = CAPSLOCK(.T.)
USE MEMBER
GO BOTTOM
DEFINE WINDOW WIN1 FROM 03,05 TO 15,67 SHADOW TITLE "Addit
ion of a member"
ACTIVATE WINDOW WIN1
CLEAR
INPUT 'How many members do you want to add : ' TO MEMNO
CLEAR
FOR I=1 TO MEMNO
   CLEAR
   SET CONFIRM ON
   APPEND BLANK
   GO BOTTOM
   STORE I TO MTEMP
   @ 0,2 SAY "Information for Member"
   @ 0,24 SAY MTEMP
   @ 1,0 SAY REPLICATE (CHR(205), 61)
   @ 8,0 SAY REPLICATE (CHR(205), 61)
   @ 2,2 SAY 'Member Name                    : ' GET NAME

   @ 3,2 SAY 'Card Number                    : ' GET CARD_NO
   @ 4,2 SAY 'Roll Number                    : ' GET ROLL_NO
   @ 5,2 SAY 'Student/Staff (S/F)            : ' GET F_OR_S
   @ 6,2 SAY 'Max. Number of books allowed : ' GET MAX_BKS
   @ 7,2 SAY 'Access to book bank (Y/N)      : ' GET AC_BNK
   READ
ENDFOR
? "Information has been appended to Members Database"
WAIT
DEACTIVATE WINDOW WIN1
CLOSE DATABASES
RETURN


*: EOF: ADDMEM.PRG
```

```
*:**********************************************************
***************
*:
*:         Program: DELBOOK.PRG
*:
*:          System: LIBRARY MANAGEMENT SYSTEM
*:          Author: HARIHARAN. S, JITESH MENON. P, KANNAN. N
*:       Copyright (c) 1994, ANONYMOUS
*:   Last modified: 09/17/94     12:48
*:
*:       Called by: BOOKPROC          (procedure in DEFMENU
.PRG)
*:
*:           Calls: FOUND()           (function  in ?)
*:
*:            Uses: NAME.DBF
*:
*:      Documented 10/20/94 at 14:45               FoxDoc
version 2.10e
*:**********************************************************
***************
*procedure for deleting books

SET TALK OFF
USE NAME
DEFINE WINDOW WIN1 FROM 03,05 TO 10,70 SHADOW TITLE "Deleti
on of a book"
ACTIVATE WINDOW WIN1
CLEAR
INPUT 'How many books do you want to delete (or 0 to quit)
: ' TO BOOKNO
IF BOOKNO = 0
   CLEAR
   RETURN
ENDIF
CLEAR
FOR I=1 TO BOOKNO
   CLEAR
   ? 'Book',I
   ? '-----'
   ?
   INPUT "Enter the Accn. Number for the book : " TO M.ACC
   ACCEPT "Are you sure (Y/N) ?  " TO M.CHOICE
   ?
   IF LOWER (M.CHOICE) = 'y'
```

```
      LOCATE FOR ACCN_NO = M.ACC
      IF FOUND() = .F.
         ? "Cannot find record with given accession number
", M.ACC
         ? "Record Not deleted, Try again !!"
         WAIT
         ?
         EXIT
      ELSE
         DELETE FOR ACCN_NO = M.ACC
         ? "Specified record deleted."
         ?
         WAIT
      ENDIF
   ENDIF
   IF LOWER (M.CHOICE) = 'n'
      ? "Record has not been deleted"
      ?
      WAIT
   ENDIF
ENDFOR
PACK
DEACTIVATE WINDOW WIN1
CLOSE DATABASES
RETURN
```

```
*:**********************************************************************
******
*:
*:        Program: LISTMEMN.PRG
*:
*:         System: LIBRARY MANAGEMENT SYSTEM
*:         Author: HARIHARAN. S, JITESH MENON. P, KANNAN. N
*:      Copyright (c) 1994, ANONYMOUS
*:  Last modified: 10/20/94     14:56
*:
*:      Called by: CASE_SM2              (procedure in DEFMENU.PRG)
*:
*:          Calls: UPPER()               (function  in ?)
*:               : EOF()                 (function  in ?)
*:               : CHR()                 (function  in ?)
*:
*:           Uses: MEMBER.DBF
*:
*:      Documented 10/20/94 at 15:27                FoxDoc  version 2.10
*:**********************************************************************
******

      SET TALK OFF
      DEFINE WINDOW SERWIN1 FROM 03,05 TO 15, 60 SHADOW TITLE "SEARCH BY
ME"
      ACTIVATE WINDOW SERWIN1
    ┌─DO WHILE .T.
    │    CLEAR
    │    @ 0,0 SAY 'Enter the name of the student or press Q to quit'
    │    ACCEPT '> 'TO MNAME
    │    CLEAR
    │  ┌─IF UPPER(MNAME) = 'Q'
    │  │    CLEAR
    v──────────EXIT
    │  └─ENDIF
    │
    │    USE MEMBER
    │
    │    LOCATE FOR UPPER(NAME) = UPPER(MNAME)
    │  ┌─IF EOF()
    │  │    ? "Cannot find student name ", MNAME
    │  │    WAIT
    ^──────────LOOP
    │  └─ENDIF
    │  ┌─DO WHILE .NOT. EOF()
    │  │    CLEAR
    │  │    @ 1,0  SAY REPLICATE (CHR(205), 53)
    │  │    @ 8,0  SAY REPLICATE (CHR(205), 53)
    │  │    @ 2,3  SAY 'Student Name          : ' GET NAME PICTURE '@T'
    │  │    @ 3,3  SAY 'Card Number           : ' GET CARD_NO PICTURE '@T'
    │  │    @ 4,3  SAY 'Roll Number           : ' GET ROLL_NO PICTURE '@T'
    │  │    @ 5,3  SAY 'Student/Staff         : ' GET F_OR_S PICTURE '@T'
    │  │    @ 6,3  SAY 'Max. Books allowed : ' GET MAX_BKS PICTURE '@T'
    │  │    @ 7,3  SAY 'Access to book bank: ' GET AC_BNK PICTURE '@T'
    │  │    CONTINUE
    │  └─ENDDO
    │    WAIT
    └─ENDDO
      DEACTIVATE WINDOW SERWIN1
      SET TALK ON
      SET HEADING ON
      CLOSE DATABASES
      CLEAR
      RETURN
```

```
*:*****************************************************************************
*:
*:         Program: LISTMEMR.PRG
*:
*:          System: LIBRARY MANAGEMENT SYSTEM
*:          Author: HARIHARAN. S. JITESH MENON. P, KANNAN. N
*:       Copyright (c) 1994, ANONYMOUS
*:   Last modified: 10/20/94     14:57
*:
*:       Called by: CASE_SM2           (procedure in DEFMENU.PRG)
*:
*:           Calls: UPPER()            (function  in ?)
*:                : EOF()              (function  in ?)
*:                : CHR()              (function  in ?)
*:
*:            Uses: MEMBER.DBF
*:
*:      Documented 10/20/94 at 15:27                   FoxDoc  version 2.10e
*:*****************************************************************************

SET TALK OFF
DEFINE WINDOW SERWIN1 FROM 03,05 TO 15, 60 SHADOW TITLE "SEARCH BY NAME"
ACTIVATE WINDOW SERWIN1
┌─DO WHILE .T.
│   CLEAR
│   @ 0,0 SAY 'Enter the Roll No. of the student or press Q to quit'
│   ACCEPT '> 'TO MNAME
│   CLEAR
│ ┌─IF UPPER(MNAME) = 'Q'
│ │     CLEAR
v─────────EXIT
│ └─ENDIF
│
│   USE MEMBER
│
│   LOCATE FOR UPPER(ROLL_NO) = UPPER(MNAME)
│ ┌─IF EOF()
│ │     ? "Cannot find student roll number ", MNAME
│ │     WAIT
└─────────LOOP
│ └─ENDIF
│ ┌─DO WHILE .NOT. EOF()
│ │     CLEAR
│ │     @ 1,0  SAY REPLICATE (CHR(205), 53)
│ │     @ 8,0  SAY REPLICATE (CHR(205), 53)
│ │     @ 2,3  SAY 'Student Name        : ' GET NAME PICTURE '@T'
│ │     @ 3,3  SAY 'Card Number         : ' GET CARD_NO PICTURE '@T'
│ │     @ 4,3  SAY 'Roll Number         : ' GET ROLL_NO PICTURE '@T'
│ │     @ 5,3  SAY 'Student/Staff       : ' GET F_OR_S PICTURE '@T'
│ │     @ 6,3  SAY 'Max. Books allowed  : ' GET MAX_BKS PICTURE '@T'
│ │     @ 7,3  SAY 'Access to book bank: ' GET AC_BNK PICTURE '@T'
│ │     CONTINUE
│ └─ENDDO
│   WAIT
└─ENDDO
DEACTIVATE WINDOW SERWIN1
SET TALK ON
SET HEADING ON
CLOSE DATABASES
CLEAR
RETURN
```

```
*:*****************************************************************************
******
*:
*:         Program: LISTMEMC.PRG
*:
*:          System: LIBRARY MANAGEMENT SYSTEM
*:          Author: HARIHARAN. S, JITESH MENON. P, KANNAN. N
*:       Copyright (c) 1994, ANONYMOUS
*:   Last modified: 10/20/94     14:57
*:
*:       Called by: CASE_SM2           (procedure in DEFMENU.PRG)
*:
*:           Calls: EOF()              (function  in ?)
*:                : CHR()              (function  in ?)
*:
*:            Uses: MEMBER.DBF
*:
*:       Documented 10/20/94 at 15:27                  FoxDoc  version 2.10e
*:*****************************************************************************
******

    SET TALK OFF
    DEFINE WINDOW SERWIN1 FROM 03,05 TO 15, 60 SHADOW TITLE "SEARCH BY N
ME"
    ACTIVATE WINDOW SERWIN1
   ┌─DO WHILE .T.
   │    CLEAR
   │    @ 0,2 SAY 'Enter the card number of the student or 0 to quit. '
   │    INPUT "> " TO M.CARD
   │    CLEAR
   │  ┌─IF M.CARD = 0
   │  │    CLEAR
   v──────────EXIT
   │  └─ENDIF
   │
   │    USE MEMBER
   │
   │    LOCATE FOR CARD_NO = M.CARD
   │  ┌─IF EOF()
   │  │    ? "Cannot find card number", M.CARD
   │  │    WAIT
   ^───────────LOOP
   │  └─ENDIF
   │  ┌─DO WHILE .NOT. EOF()
   │  │    CLEAR
   │  │    @ 1,0  SAY REPLICATE (CHR(205), 53)
   │  │    @ 8,0  SAY REPLICATE (CHR(205), 53)
   │  │    @ 2,3  SAY 'Student Name        : ' GET NAME PICTURE '@T'
   │  │    @ 3,3  SAY 'Card Number         : ' GET CARD_NO PICTURE '@T'
   │  │    @ 4,3  SAY 'Roll Number         : ' GET ROLL_NO PICTURE '@T'
   │  │    @ 5,3  SAY 'Student/Staff       : ' GET F_OR_S PICTURE '@T'
   │  │    @ 6,3  SAY 'Max. Books allowed : ' GET MAX_BKS PICTURE '@T'
   │  │    @ 7,3  SAY 'Access to book bank: ' GET AC_BNK PICTURE '@T'
   │  │    CONTINUE
   │  └─ENDDO
   │    WAIT
   └─ENDDO
    DEACTIVATE WINDOW WIN1
    SET TALK ON
    SET HEADING ON
    CLOSE DATABASES
    CLEAR
    RETURN
```

```
*:*****************************************************************************
******
*:
*:        Program: MAINMENU.PRG
*:
*:         System: LIBRARY MANAGEMENT SYSTEM
*:         Author: HARIHARAN. S, JITESH MENON. P, KANNAN. N
*:      Copyright (c) 1994, ANONYMOUS
*:  Last modified: 10/20/94     14:59
*:
*:          Calls: BORBOOK.PRG
*:               : RETBOOK.PRG
*:               : DEFMENU.PRG
*:
*:      Documented 10/20/94 at 15:27                        FoxDoc  version 2.10e
*:*****************************************************************************
******

     *Main menu
     CLEAR
     CLEAR ALL
     SET TALK OFF
   ┌─DO WHILE .T.
   │    DEFINE WINDOW MAIN1 FROM 05,25 TO 16, 56 TITLE
   │    " MAIN MENU " PANEL SHADOW
   │    DEFINE WINDOW MAIN2 FROM 00,01 TO 02, 78 DOUBLE SHADOW
   │    DEFINE WINDOW MAIN3 FROM 20,01 TO 22, 78 SHADOW
   │    ACTIVATE WINDOW MAIN2
   │    @ 0,27 SAY 'LIBRARY MANAGEMENT SYSTEM'
   │    ACTIVATE WINDOW MAIN3
   │    @ 0,8  SAY 'Case study lab by Hariharan. S,
   │    JITESH MENON. P AND KANNAN. N'
   │    ACTIVATE WINDOW MAIN1
   │    @ 2, 8  SAY '1. Lend'
   │    @ 3, 8  SAY '2. Receive'
   │    @ 4, 8  SAY '3. Control menu'
   │    @ 5, 8  SAY '4. Help'
   │    @ 6, 8  SAY '5. Exit'
   │    ?
   │    WAIT '  Enter your choice : ' TO CHOICE
   │
   │  ┌─DO CASE
   │  ├─CASE CHOICE = '1'
   │  │     DO BORBOOK
   │  ├─CASE CHOICE = '2'
   │  │     DO RETBOOK
   │  ├─CASE CHOICE = '3'
   │  │     DO DEFMENU
   │  ├─CASE CHOICE = '4'
   │  │     DO DEFMENU
   │  ├─CASE CHOICE = '5'
   │  │     DEACTIVATE WINDOW MAIN1
   │  │     DEACTIVATE WINDOW MAIN2
   │  │     DEACTIVATE WINDOW MAIN3
   v──┴─────EXIT
   │  ├─OTHERWISE
   │  │     DEFINE WINDOW POP1 FROM 3, 35 TO 8, 70 SHADOW
   │  │     ACTIVATE WINDOW POP1
   │  │     @ 1, 5 SAY 'Wrong Choice entered'
```

```
 │  │     WAIT
 │  │     DEACTIVATE WINDOW POP1
 │  └─ENDCASE
 │     DEACTIVATE WINDOW MAIN1
 │     DEACTIVATE WINDOW MAIN2
 │     DEACTIVATE WINDOW MAIN3
 └─ENDDO
    SET TALK ON
    *: EOF: MAINMENU.PRG
```

```
*:*****************************************************************************
******
*:
*:         Program: RETBOOK.PRG
*:
*:          System: LIBRARY MANAGEMENT SYSTEM
*:          Author: HARIHARAN. S, JITESH MENON. P, KANNAN. N
*:       Copyright (c) 1994, ANONYMOUS
*:   Last modified: 10/20/94     15:02
*:
*:       Called by: MAINMENU.PRG
*:
*:           Calls: CAPSLOCK()          (function  in ?)
*:                : UPPER()             (function  in ?)
*:                : EOF()               (function  in ?)
*:                : DATE()              (function  in ?)
*:                : CTOD()              (function  in ?)
*:                : CHR()               (function  in ?)
*:
*:            Uses: MEMBER.DBF
*:                : NAME.DBF
*:                : CIRC.DBF
*:
*:       Documented 10/20/94 at 15:27               FoxDoc  version 2.10
*:*****************************************************************************
******

    SET TALK OFF
    CAPSTATUS = CAPSLOCK(.T.)
    DEFINE WINDOW SERWIN1 FROM 03,05 TO 15, 60 SHADOW TITLE "Return a b
k"
    ACTIVATE WINDOW SERWIN1
    *DO WHILE .T.
    *-----------------------------------------------------------------------
    *Searches for the Roll Number in the member database
    CLEAR
    @ 0,0 SAY 'Enter the Roll No. of the student or press Q to quit'
    ACCEPT '> 'TO M.ROL
    CLEAR
  ┌─IF UPPER(M.ROL) = 'Q'
  │    CLEAR
  │    DEACTIVATE WINDOW SERWIN1
<─────RETURN
  └─ENDIF

    USE MEMBER

    LOCATE FOR UPPER(ROLL_NO) = UPPER(M.ROL)
```

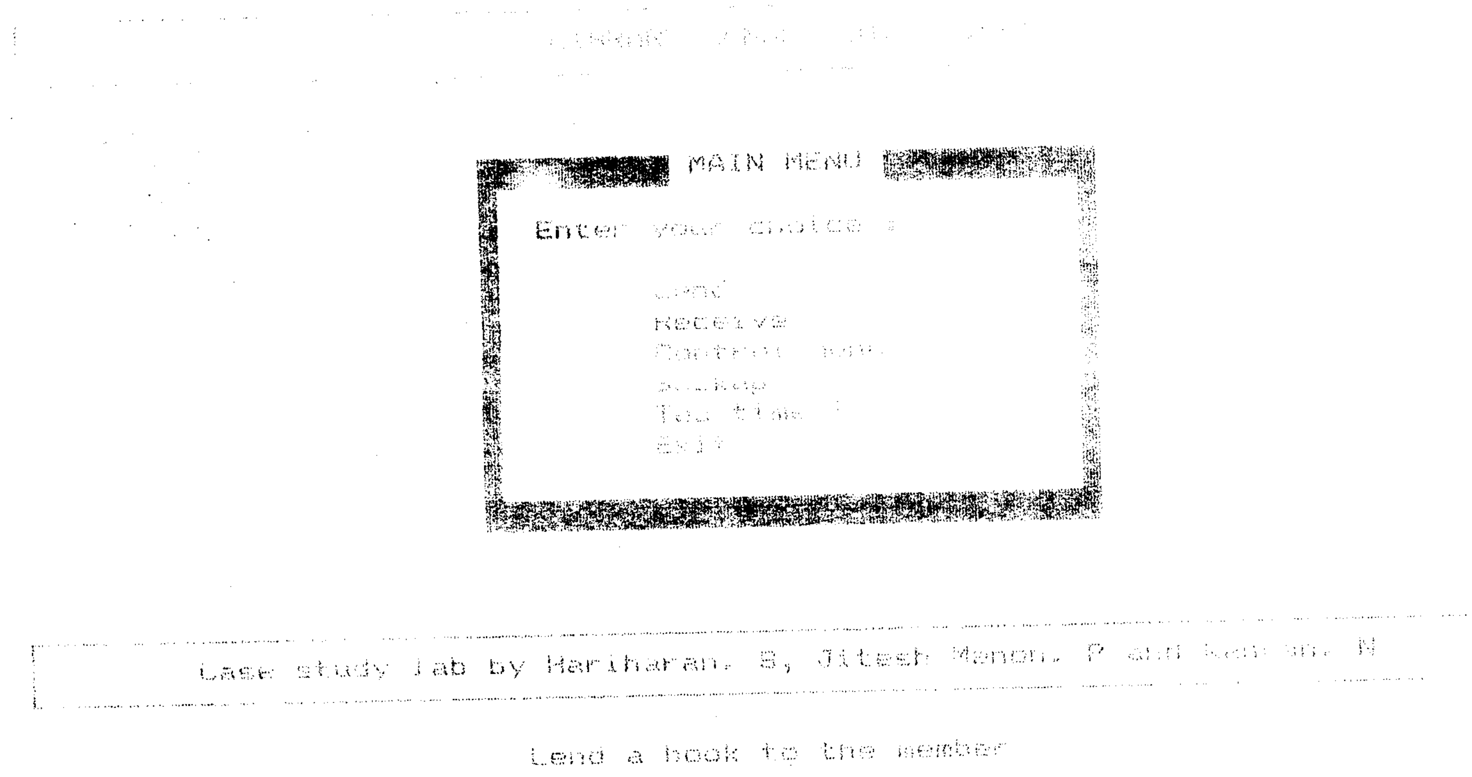
```
   ┌─IF EOF()
   │     ? "Cannot find student roll number ", M.ROL
   │     WAIT
   │     DEACTIVATE WINDOW SERWIN1
 <───────RETURN
   └─ENDIF
     * STORE MAX_BKS TO max
     *-----------------------------------------------------------------------
     *-----------------------------------------------------------------------
     *Searches for Accn. No in the book database

     @ 1,0 SAY 'Enter the Accn. No. of the book or press 0 to quit'
     INPUT '> 'TO MNO
     CLEAR
   ┌─IF MNO = 0
   │     CLEAR
   │     DEACTIVATE WINDOW SERWIN1
 <───────RETURN
   └─ENDIF

     USE NAME

     LOCATE FOR ACCN_NO = MNO
   ┌─IF EOF()
   │     ? "Cannot find given Accn. No. of the book", MNO
   │     WAIT
   │     DEACTIVATE WINDOW SERWIN1
 <───────RETURN
   └─ENDIF
     *-----------------------------------------------------------------------
     *Searches the circulation database to check whether there is an
     *existing record for this roll number

     USE CIRC
     LOCATE FOR ROLL_NO = M.ROL
   ┌─IF EOF()
   │     CLEAR
   │     @ 1, 1 SAY 'This person has not borrowed any book -  '+M.ROL
   │     WAIT
   │     DEACTIVATE WINDOW SERWIN1
 <───────RETURN
   └─ENDIF

   ┌─DO CASE
   ├─CASE ACCN_NO_1 = MNO
   │     REPLACE ACCN_NO_1 WITH 0
   │     DAYS = (DATE() - CTOD(DUE_DATE_1))
   │   ┌─IF DAYS > 0
   │   │     T_FINE = FINE + DAYS * 0.50
   │   │     REPLACE FINE WITH T_FINE
   │   └─ENDIF
   │     REPLACE DUE_DATE_1 WITH ' '
   ├─CASE ACCN_NO_2 = MNO
   │     REPLACE ACCN_NO_2 WITH 0
   │     DAYS = (DATE()   CTOD(DUE_DATE_2))
   │   ┌─IF DAYS > 0
   │   │     T_FINE = FINE + DAYS * 0.50
   │   │     REPLACE FINE WITH T_FINE
   │   └─ENDIF
   │     REPLACE DUE_DATE_2 WITH ' '
   ├─CASE ACCN_NO_3 = MNO
   REPLACE ACCN_NO_3 WITH 0
   DAYS = (DATE() - CTOD(DUE_DATE_3))
   IF DAYS > 0
      T_FINE = FINE + DAYS * 0.50
      REPLACE FINE WITH T_FINE
   ENDIF
   REPLACE DUE_DATE_3 WITH ' '
CASE ACCN_NO_4 = MNO
   REPLACE ACCN_NO_4 WITH 0
   DAYS = (DATE() - CTOD(DUE_DATE_4))
   IF DAYS > 0
      T_FINE = FINE + DAYS * 0.50
      REPLACE FINE WITH T_FINE
   ENDIF
   REPLACE DUE_DATE_4 WITH ' '
CASE ACCN_NO_5 = MNO
   REPLACE ACCN_NO_1 WITH 0
   DAYS = (DATE() - CTOD(DUE_DATE_5))
   IF DAYS > 0
      T_FINE = FINE + DAYS * 0.50
      REPLACE FINE WITH T_FINE
   ENDIF
   REPLACE DUE_DATE_5 WITH ' '
OTHERWISE
   CLEAR
   @ 1,1 SAY 'The book with the Accn. No'
   @ 1,28 SAY MNO
   @ 2,1 SAY 'has not been borrowed by Roll No. '+M.ROL
   ?
   WAIT
   DEACTIVATE WINDOW SERWIN1
   RETURN
ENDCASE
TOT = TOT_BOOKS - 1
REPLACE TOT_BOOKS WITH TOT

IF FINE > 0
   CLEAR
   @ 0,0  SAY REPLICATE (CHR(205), 52)
   @ 6,0  SAY REPLICATE (CHR(205), 52)
   @ 1,7  SAY 'Roll Number            : '+M.ROL
   @ 2,7  SAY 'Accn. No               : '
   @ 2,30 SAY MNO PICT '99999'
   @ 3,7  SAY 'Lag period             : '
   @ 3,30 SAY DAYS PICT '999.99'
   @ 3,34 SAY 'days'
   @ 4,7  SAY 'Fine  (0.5/day)        : '
   @ 4,30 SAY (DAYS*0.50) PICT '999.99'
   @ 5,7  SAY 'Total Fine             : '
   @ 5,30 SAY FINE PICT '999.99'
   ?
   ACCEPT 'Collect fine now (Y/N) ? ' TO CHOICE
ENDIF

DO CASE
CASE UPPER(CHOICE) = 'Y'
   REPLACE FINE WITH 0
   CLEAR
   @ 1,1 SAY 'Collect cash and issue receipt'
CASE UPPER(CHOICE) = 'N'
   CLEAR
```

```
│   @ 1,1 SAY ’Inform member to pay cash as soon as possible.’
└─ENDCASE
  ?
  WAIT

  USE NAME

  LOCATE FOR ACCN_NO = MNO
  REPLACE STATUS WITH ’Y’

  DEACTIVATE WINDOW SERWIN1
  CLEAR
  CLOSE DATABASES
  SET TALK ON

  *: EOF: RETBOOK.PRG


*:*******************************************************************
******
*:
*: Procedure file: SORTMEM.PRG
*:
*:         System: LIBRARY MANAGEMENT SYSTEM
*:         Author: HARIHARAN. S, JITESH MENON. P, KANNAN. N
*:      Copyright (c) 1994, ANONYMOUS
*:  Last modified: 10/20/94     15:04
*:
*:         Set by: MEMPROC              (procedure in DEFMENU.PRG)
*:
*:          Calls: PROMPT()             (function  in ?)
*:               : CASE_SM1             (procedure in DEFMENU.PRG)
*:
*:           Uses: NAME.DBF
*:
*:     Documented 10/20/94 at 15:27               FoxDoc  version 2.10e
*:*******************************************************************
******
    *Procedure for sorting members
    CLEAR
    SET TALK OFF
    USE NAME

    DEFINE POPUP SM_2 FROM 6,15 COLOR SCHEME 2 TITLE "Sort" SHADOW
    DEFINE BAR 1 OF SM_2 PROMPT "Sort by \<Name"
    DEFINE BAR 2 OF SM_2 PROMPT "Sort by \<Roll No."
    DEFINE BAR 3 OF SM_2 PROMPT "Sort by \<Student/Staff"
    DEFINE BAR 4 OF SM_2 PROMPT "Sort by \<Card No."
    DEFINE BAR 5 OF SM_2 PROMPT "Sort by \<Total Books"
    ON SELECTION POPUP SM_2 DO CASE_SM1 WITH PROMPT()

    ACTIVATE POPUP SM_2

    *----------------------------------------------------------------

*!*******************************************************************
******
*!
*!      Procedure: CASE_SM2
*!
*!      Called by: SUB_FOR_LIST          (procedure in DEFMENU.PRG)
*!
*!          Calls: LISTMEMN.PRG
*!                  : LISTMEMR.PRG
*!                  : LISTMEMC.PRG
*!
*!              Uses: MEMBER.DBF
*!                  : TEMP1.DBF
*!                  : TEM1P1.DBF
*!
*!**************************************************************************
    PROCEDURE CASE_SM2
    *Procedure for case structure for submenu
    PARAMETER  MPROMPT

   ┌─DO CASE
   ├─CASE MPROMPT = 'Sort by Book name'
   │    USE MEMBER
   │    SORT TO TEMP1 ON BOOK_NAME
   │    CLOSE DATABASES
   │    COPY FILE TEMP1.DBF TO MEMBER.DBF
   ├─CASE MPROMPT = 'Sort by Author'
   │    USE MEMBER
   │    SORT TO TEMP1 ON BOOK_NAME
   │    CLOSE DATABASES
   │    COPY FILE TEMP1.DBF TO MEMBER.DBF
   ├─CASE MPROMPT = 'Sort by Accn. No.'
   │    USE MEMBER
   │    SORT TO TEMP1 ON BOOK_NAME
   │    CLOSE DATABASES
   │    COPY FILE TEMP1.DBF TO MEMBER.DBF
   ├─CASE MPROMPT = 'Sort by Subject'
   │    USE MEMBER
   │    SORT TO TEMP1 ON BOOK_NAME
   │    CLOSE DATABASES
   │    COPY FILE TEM1P1.DBF TO MEMBER.DBF
   ├─CASE MPROMPT = 'Sort by Total books
   └─ENDCASE
     RETURN


     *: EOF: SORTMEM.PRG
```

```
*:*****************************************************************************
******
*:
*: Procedure file: SORTBOOK.PRG
*:
*:         System: LIBRARY MANAGEMENT SYSTEM
*:         Author: HARIHARAN. S, JITESH MENON. P, KANNAN. N
*:      Copyright (c) 1994, ANONYMOUS
*:  Last modified: 10/20/94     14:53
*:
*:         Set by: BOOKPROC               (procedure in DEFMENU.PRG)
*:
*:          Calls: PROMPT()               (function  in ?)
*:               : CASE_SM1               (procedure in DEFMENU.PRG)
*:
*:           Uses: NAME.DBF
*:
*:      Documented 10/20/94 at 15:27                FoxDoc  version 2.10
*:*****************************************************************************
******
    *Procedure for sorting a book
    CLEAR
    SET TALK OFF
    USE NAME

    DEFINE POPUP SM_1 FROM 6,15 COLOR SCHEME 2 TITLE "Sort" SHADOW
    DEFINE BAR 1 OF SM_1 PROMPT "Sort by \<Book name"
    DEFINE BAR 2 OF SM_1 PROMPT "Sort by \<Author"
    DEFINE BAR 3 OF SM_1 PROMPT "Sort by Acc\<n. No."
    DEFINE BAR 4 OF SM_1 PROMPT "Sort by \<Subject"
    ON SELECTION POPUP SM_1 DO CASE_SM1 WITH PROMPT()
    ACTIVATE POPUP SM_1

    *--------------------------------------------------------------------

*!*****************************************************************************
******
*!
*!      Procedure: CASE_SM1
*!
*!      Called by: SUB_FOR_SER            (procedure in DEFMENU.PRG)
*!               : SORTBOOK.PRG
*!               : SORTMEM.PRG
*!
*!          Calls: SERBOOKN.PRG
*!               : SERBOOKA.PRG
*!
*!           Uses: NAME.DBF
*!               : TEMP.DBF
*!
*!*****************************************************************************
******
    PROCEDURE CASE_SM1
    *Procedure for case structure for submenu
    PARAMETER  MPROMPT

  ┌─DO CASE
  ├─CASE MPROMPT = 'Sort by Book name'
  │     USE NAME
  │     SORT TO TEMP ON BOOK_NAME
  │     CLOSE DATABASES
```

```
       │   COPY FILE TEMP.DBF TO NAME.DBF
       ├─CASE MPROMPT = 'Sort by Author'
       │   USE NAME
       │   SORT TO TEMP ON AUTHOR
       │   CLOSE DATABASES
       │   COPY FILE TEMP.DBF TO NAME.DBF
       ├─CASE MPROMPT = 'Sort by Accn. No.'
       │   USE NAME
       │   SORT TO TEMP ON ACCN_NO
       │   CLOSE DATABASES
       │   COPY FILE TEMP.DBF TO NAME.DBF
       ├─CASE MPROMPT = 'Sort by Subject'
       │   USE NAME
       │   SORT TO TEMP ON SUBJECT
       │   CLOSE DATABASES
       │   COPY FILE TEMP.DBF TO NAME.DBF
       └─ENDCASE
         RETURN


         *: EOF: SORTBOOK.PRG

   *:***************************************************************************
******
   *:
   *:         Program: SERBOOKN.PRG
   *:
   *:          System: LIBRARY MANAGEMENT SYSTEM
   *:          Author: HARIHARAN. S, JITESH MENON. P, KANNAN. N
   *:       Copyright (c) 1994, ANONYMOUS
   *:   Last modified: 10/20/94     14:56
   *:
   *:       Called by: CASE_SM1             (procedure in DEFMENU.PRG)
   *:
   *:           Calls: UPPER()              (function  in ?)
   *:                : EOF()                (function  in ?)
   *:                : CHR()                (function  in ?)
   *:
   *:            Uses: NAME.DBF
   *:
   *:      Documented 10/20/94 at 15:27                   FoxDoc  version 2.10
   *:***************************************************************************
******

         SET TALK OFF
         SET HEADING OFF
         PRIVATE MNAME
         DEFINE WINDOW SERWIN1 FROM 03,05 TO 20, 65 FLOAT
         SHADOW TITLE "Search by Author"
         ACTIVATE WINDOW SERWIN1
       ┌─DO WHILE .T.
       │   CLEAR
       │   @ 0,0 SAY 'Enter the name of the book or press Q to quit. '
       │   ACCEPT '> ' TO MNAME
       │   CLEAR
       │ ┌─IF UPPER(MNAME) = 'Q'
       │ │   CLEAR
       v─────EXIT
       │ │   DEACTIVATE WINDOW SERWIN1
       │ └─ENDIF
       │
       │   USE NAME
   │
   │    LOCATE FOR UPPER(BOOK_NAME) = UPPER(MNAME)
   │ ┌─IF EOF()
   │ │     ? " Cannot find the book ", MNAME
   │ │     WAIT
   └───────LOOP
   │ └─ENDIF
   │ ┌─DO WHILE .NOT. EOF()
   │ │     CLEAR
   │ │     @ 0,0  SAY REPLICATE (CHR(205), 59)
   │ │     @ 14,0 SAY REPLICATE (CHR(205), 59)
   │ │     @ 1,4  SAY 'Book Name : ' GET BOOK_NAME PICTURE '@T'
   │ │     @ 2,4  SAY 'Author    : ' GET AUTHOR PICTURE '@T'
   │ │     @ 3,4  SAY 'Volume    : ' GET VOLUME PICTURE '@T'
   │ │     @ 4,4  SAY 'Edition   : ' GET EDITION PICTURE '@T'
   │ │     @ 5,4  SAY 'Copies    : ' GET COPIES PICTURE '@T'
   │ │     @ 6,4  SAY 'Text/Ref. : ' GET T_OR_REF PICTURE '@T'
   │ │     @ 7,4  SAY 'Bank/Gen. : ' GET BNK_OR_GNR PICTURE '@T'
   │ │     @ 8,4 SAY 'Call No.  : ' GET CALL_NO PICTURE '@T'
   │ │     @ 9,4 SAY 'Accn. No. : ' GET ACCN_NO PICTURE '@T'
   │ │     @ 10,4 SAY 'Publisher : ' GET PUBLISHER PICTURE '@T'
   │ │     @ 11,4 SAY 'Price     : ' GET PRICE PICTURE '@T'
   │ │     @ 12,4 SAY 'Agents    : ' GET AGENTS PICTURE '@T'
   │ │     @ 13,4 SAY 'Bill No.  : ' GET BILL_NO PICTURE '@T'
   │ │     CONTINUE
   │ └─ENDDO
   │     ?
   │     WAIT
   └─ENDDO
     DEACTIVATE WINDOW SERWIN1
     SET TALK ON
     SET HEADING ON
     CLOSE DATABASES
     CLEAR
     RETURN

     *: EOF: SERBOOKN.PRG
*:*********************************************************************
******
*:
*:        Program: SERBOOKA.PRG
*:
*:         System: LIBRARY MANAGEMENT SYSTEM
*:         Author: HARIHARAN. S, JITESH MENON. P, KANNAN. N
*:      Copyright (c) 1994, ANONYMOUS
*:  Last modified: 10/20/94     14:55
*:
*:      Called by: CASE_SM1          (procedure in DEFMENU.PRG)
*:
*:          Calls: UPPER()           (function  in ?)
*:               : EOF()             (function  in ?)
*:               : CHR()             (function  in ?)
*:
*:           Uses: NAME.DBF
*:
*:     Documented 10/20/94 at 15:27               FoxDoc  version 2.10
*:*********************************************************************
******

     SET TALK OFF
```

```
      SET HEADING OFF
      PRIVATE MNAME
      DEFINE WINDOW SERWIN1 FROM 03,05 TO 20, 65 FLOAT SHADOW TITLE "Search
by Author"
      ACTIVATE WINDOW SERWIN1
   ┌──DO WHILE .T.
   │     CLEAR
   │     @ 0,0 SAY 'Enter the name of the Author or press Q to quit. '
   │     ACCEPT '> ' TO MNAME
   │     CLEAR
   │  ┌──IF UPPER(MNAME) = 'Q'
   │  │     CLEAR
   v──────────EXIT
   │  │     DEACTIVATE WINDOW SERWIN1
   │  └──ENDIF
   │
   │     USE NAME
   │
   │     LOCATE FOR UPPER(AUTHOR) = UPPER(MNAME)
   │  ┌──IF EOF()
   │  │     ? " Cannot find the author ", MNAME
   │  │     WAIT
   ^──────────LOOP
   │  └──ENDIF
   │  ┌──DO WHILE .NOT. EOF()
   │  │     CLEAR
   │  │     @ 0,0  SAY REPLICATE (CHR(205), 59)
   │  │     @ 14,0 SAY REPLICATE (CHR(205), 59)
   │  │     @ 1,4  SAY 'Book Name : ' GET BOOK_NAME PICTURE '@T'
   │  │     @ 2,4  SAY 'Author    : ' GET AUTHOR PICTURE '@T'
   │  │     @ 3,4  SAY 'Volume    : ' GET VOLUME PICTURE '@T'
   │  │     @ 4,4  SAY 'Edition   : ' GET EDITION PICTURE '@T'
   │  │     @ 5,4  SAY 'Copies    : ' GET COPIES PICTURE '@T'
   │  │     @ 6,4  SAY 'Text/Ref. : ' GET T_OR_REF PICTURE '@T'
   │  │     @ 7,4  SAY 'Bank/Gen. : ' GET BNK_OR_GNR PICTURE '@T'
   │  │     @ 8,4 SAY 'Call No.   : ' GET CALL_NO PICTURE '@T'
   │  │     @ 9,4 SAY 'Accn. No.  : ' GET ACCN_NO PICTURE '@T'
   │  │     @ 10,4 SAY 'Publisher : ' GET PUBLISHER PICTURE '@T'
   │  │     @ 11,4 SAY 'Price     : ' GET PRICE PICTURE '@T'
   │  │     @ 12,4 SAY 'Agents    : ' GET AGENTS PICTURE '@T'
   │  │     @ 13,4 SAY 'Bill No.  : ' GET BILL_NO PICTURE '@T'
   │  │     CONTINUE
   │  └──ENDDO
   │     ?
   │     WAIT
   └──ENDDO
      DEACTIVATE WINDOW SERWIN1
      SET TALK ON
      SET HEADING ON
      CLOSE DATABASES
      CLEAR
      RETURN

      *: EOF: SERBOOKA.PRG
```



LIBRARY MANAGEMENT SYSTEM

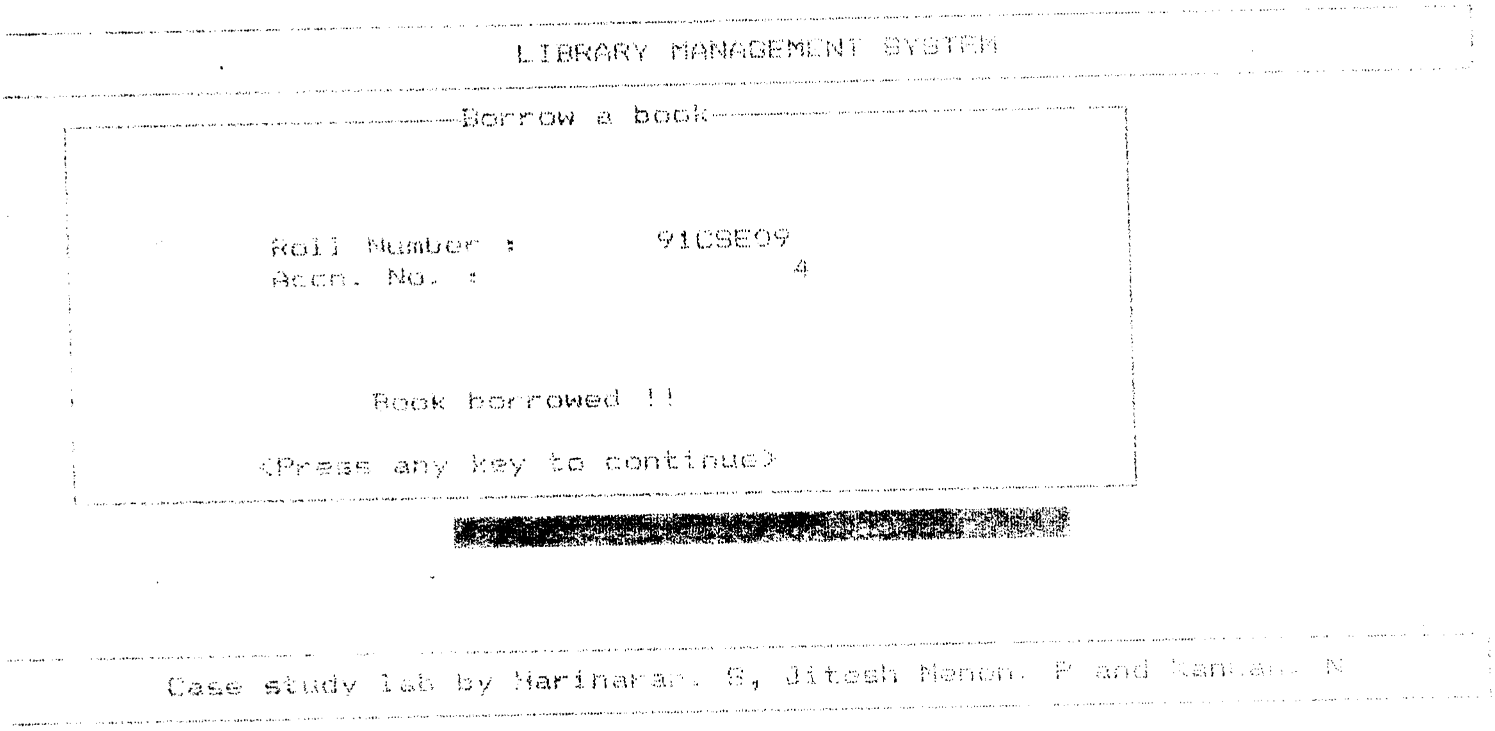
MAIN MENU

Enter your choice :

Lend
Receive
Control room
Backup
Tea time !
Exit

Case study lab by Hariharan. S, Jitesh Menon. P and Kannan. N

Lend a book to the member

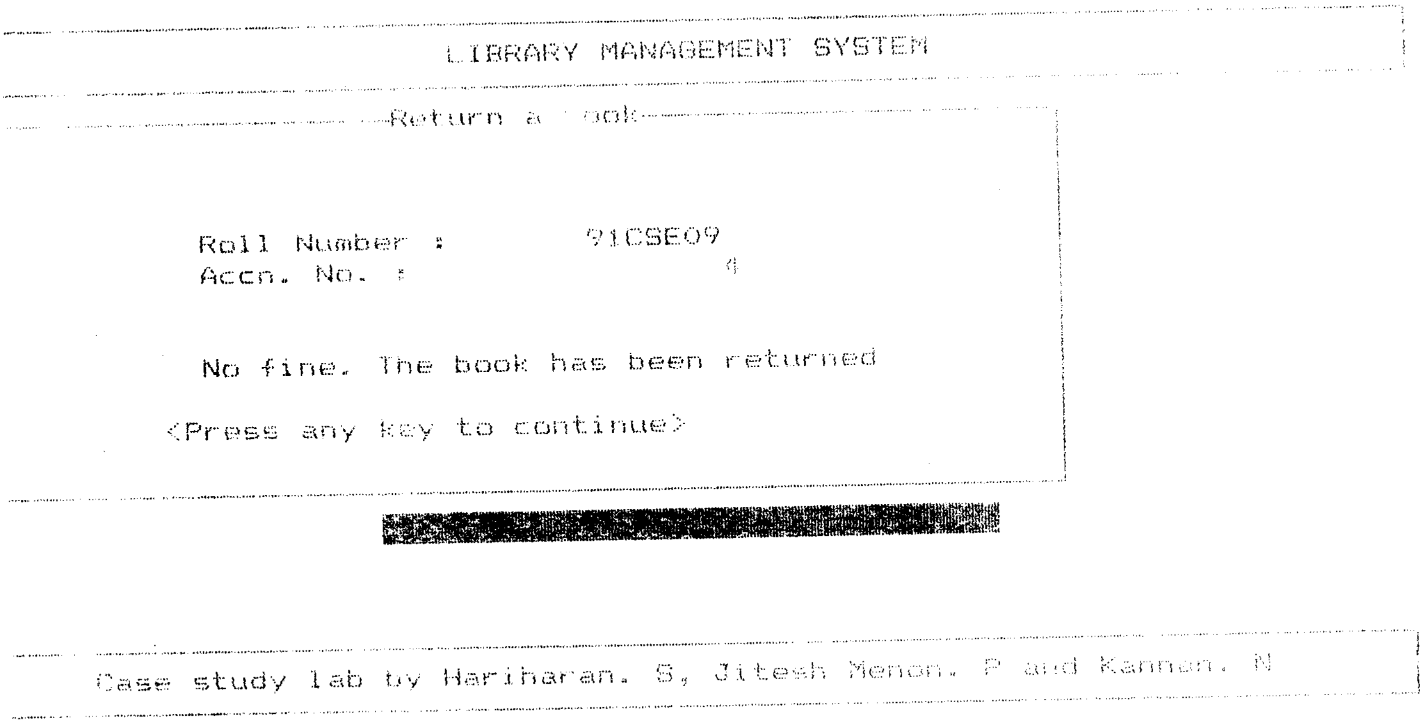


LIBRARY MANAGEMENT SYSTEM

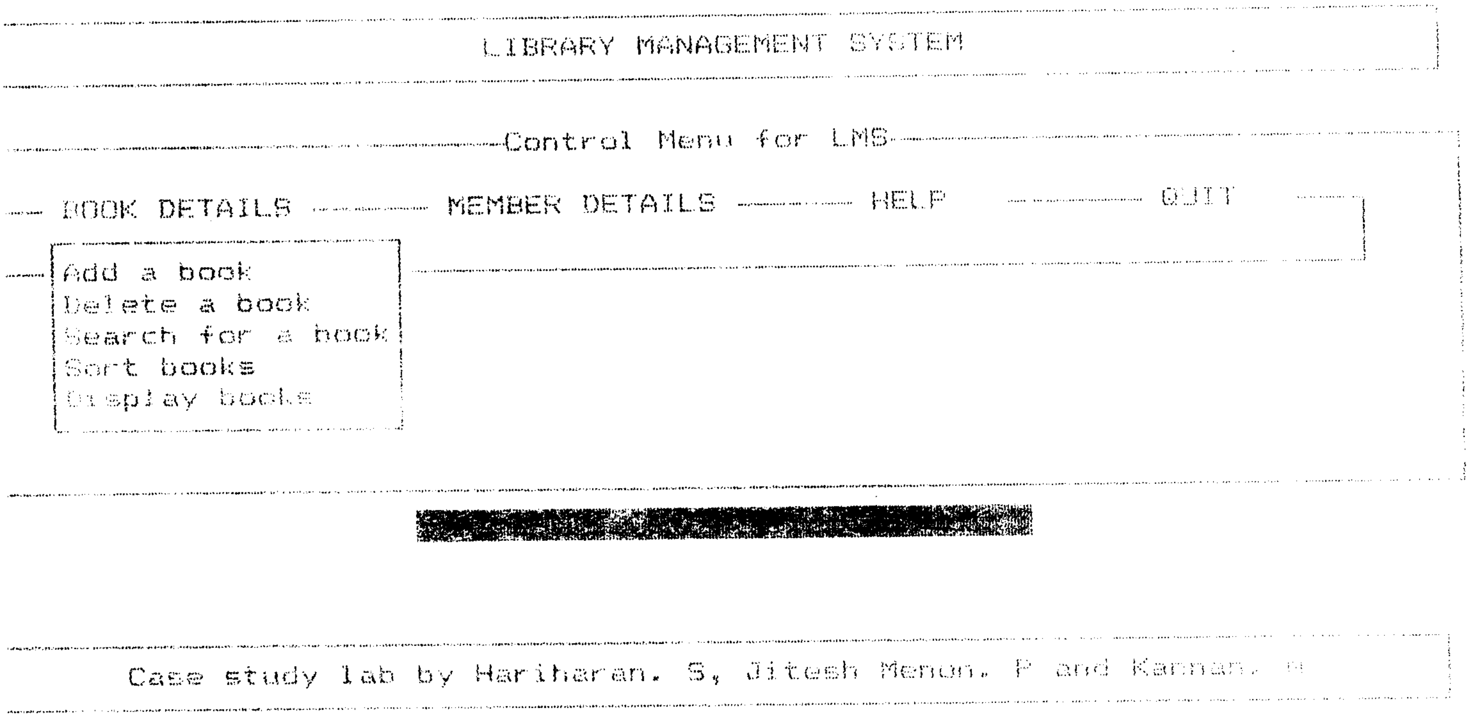
Borrow a book

```
Roll Number :          91CSE09
Accn. No. :                  4


Book borrowed !!

<Press any key to continue>
```

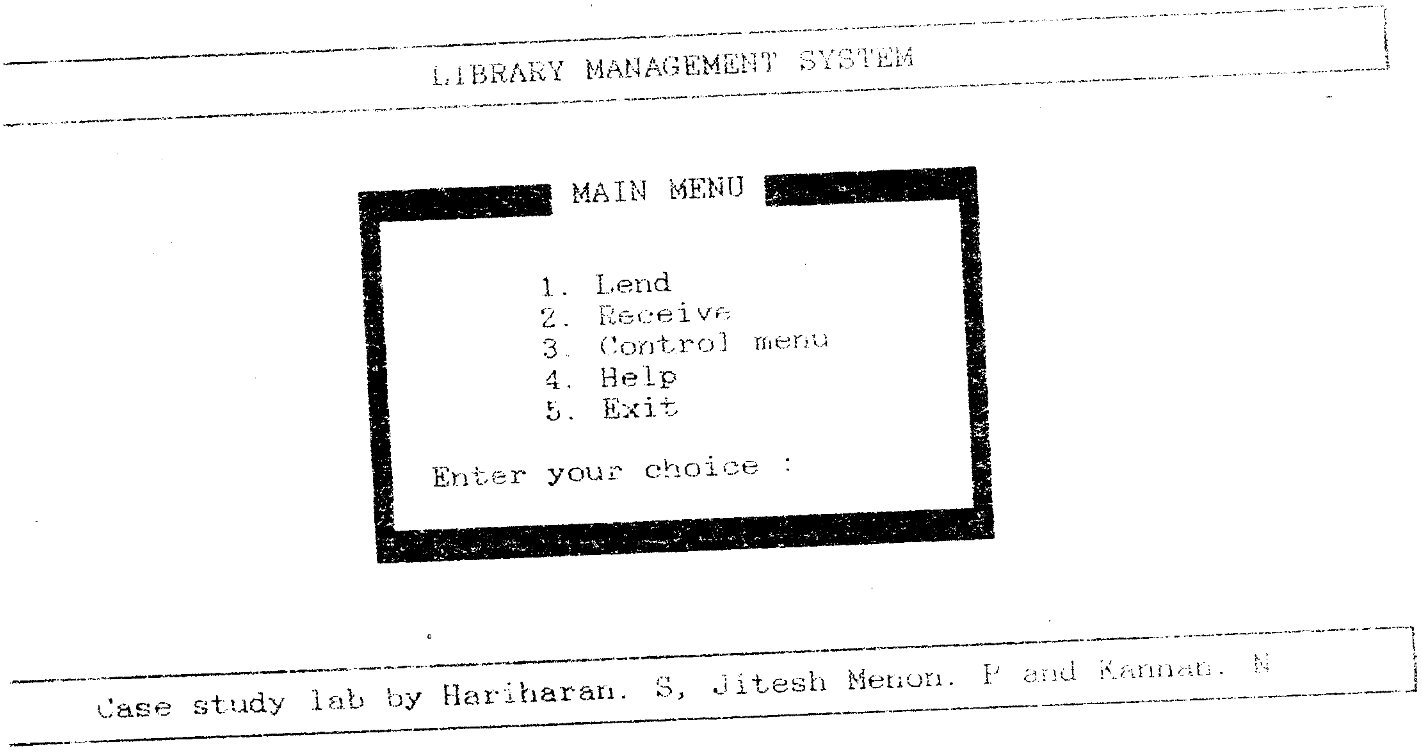
Case study lab by Hariharan. S, Jitesh Menon. P and Kannan. N



```
LIBRARY MANAGEMENT SYSTEM

Return a book

    Roll Number :         91CSE09
    Accn. No. :                 4

    No fine. The book has been returned

  <Press any key to continue>

Case study lab by Hariharan. S, Jitesh Menon. P and Kannan. N
```

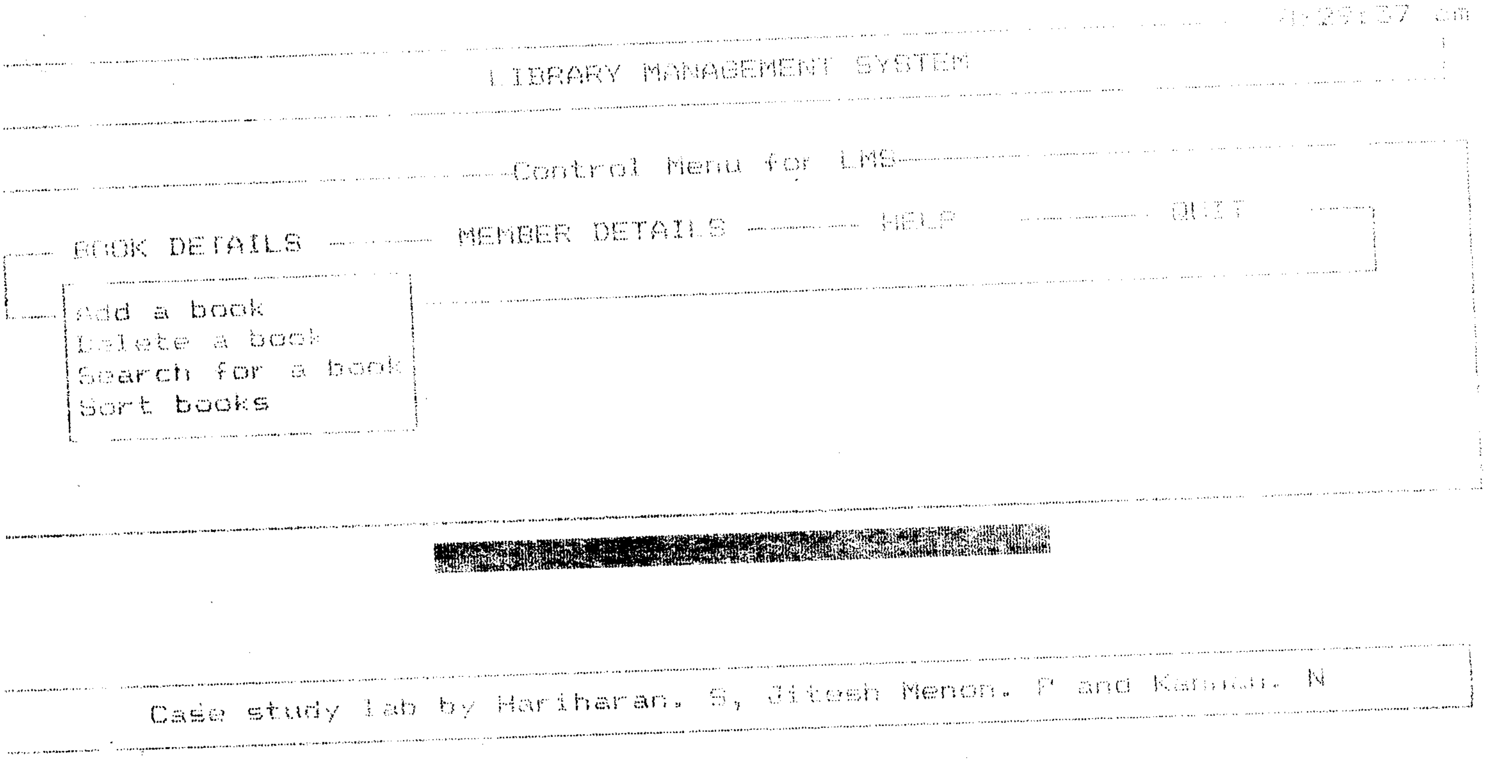


```
LIBRARY MANAGEMENT SYSTEM

Control Menu for LMS

BOOK DETAILS   MEMBER DETAILS   HELP   QUIT
 Add a book
 Delete a book
 Search for a book
 Sort books
 Display books

Case study lab by Hariharan. S, Jitesh Menon. P and Kannan. N
```

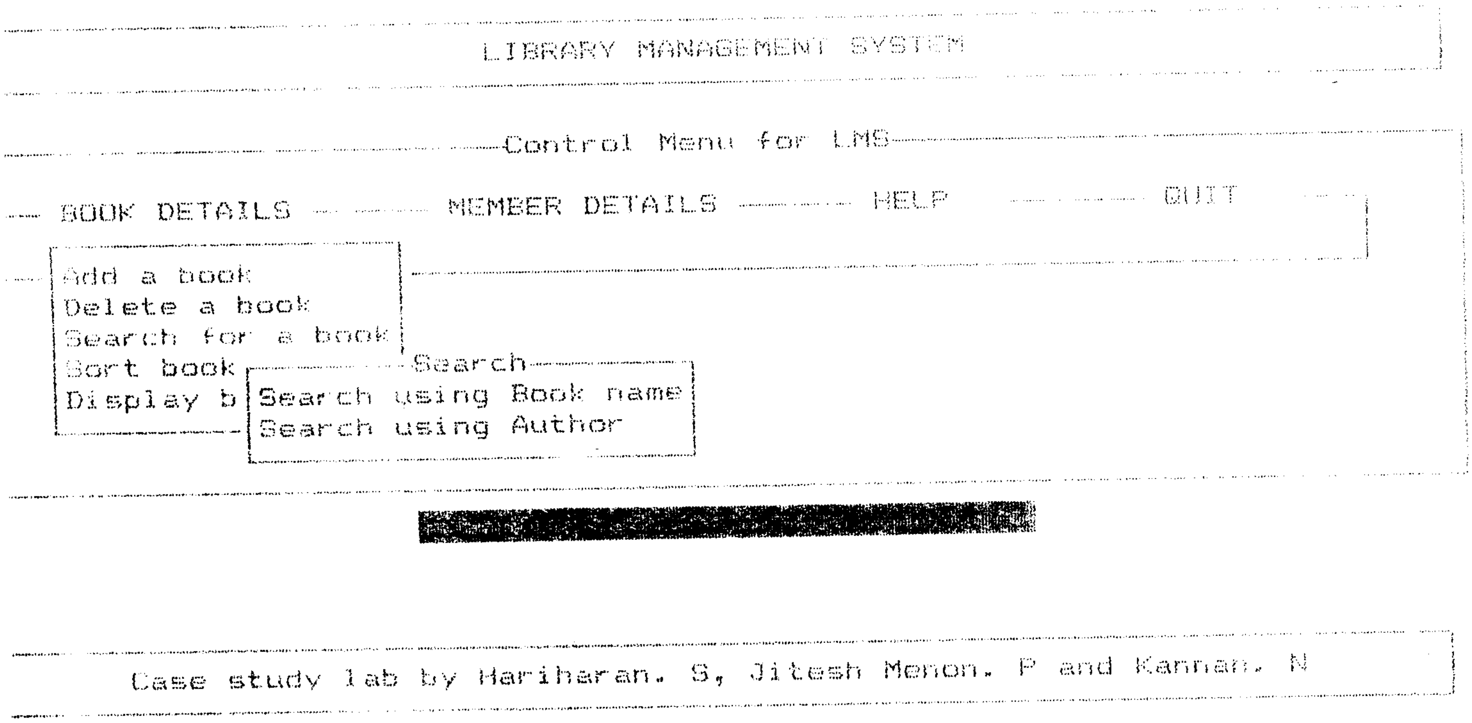


Enter your choice :

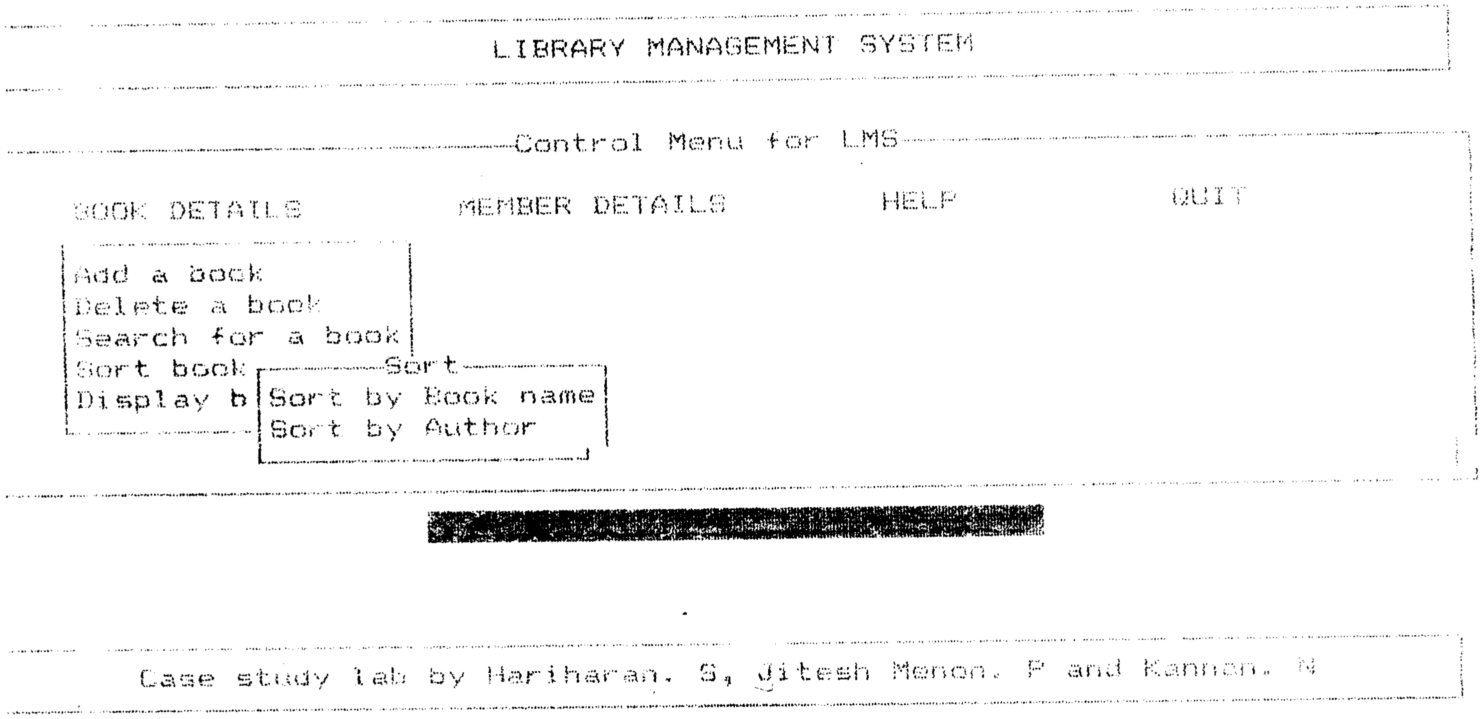
Case study lab by Hariharan. S, Jitesh Menon. P and Kannan. N



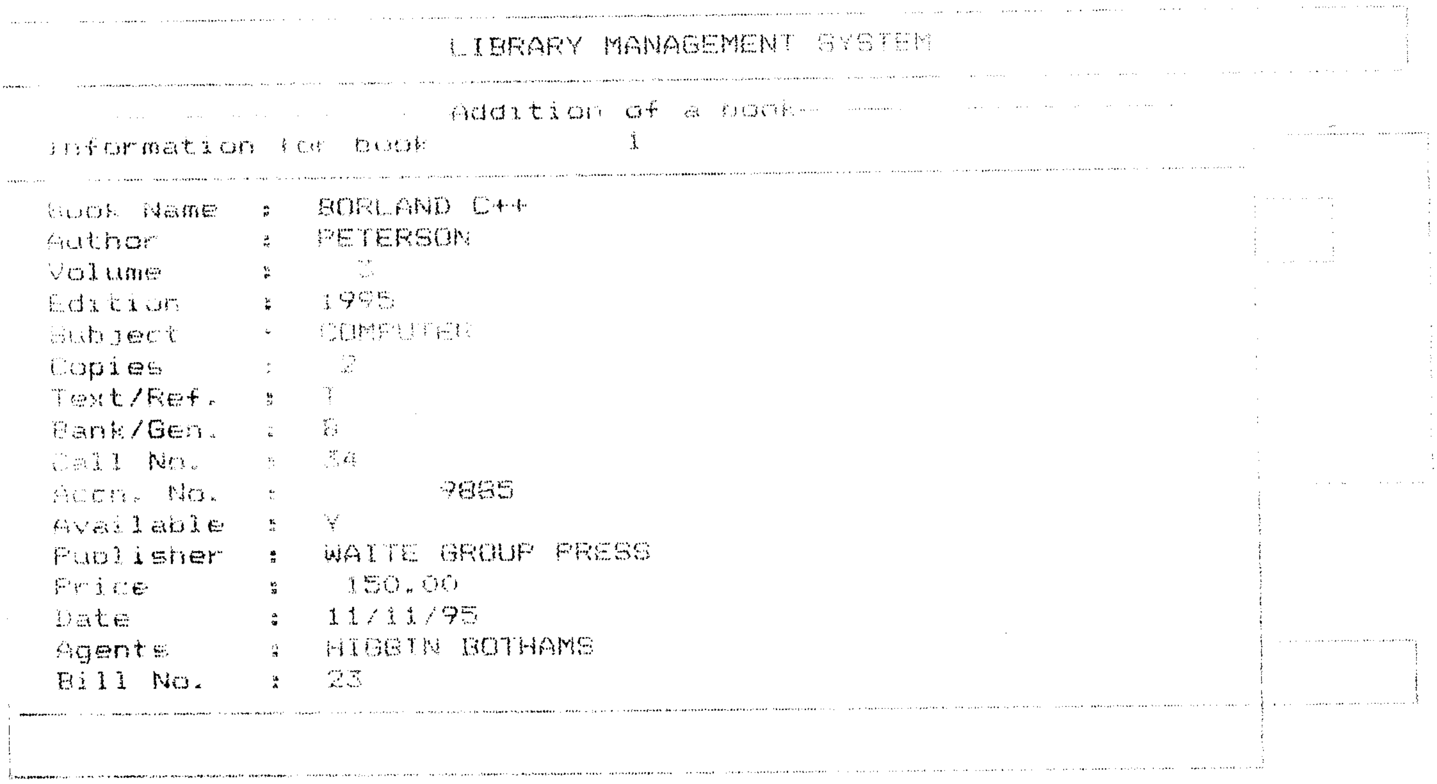
Case study lab by Hariharan. S, Jitesh Menon. P and Kannan. N



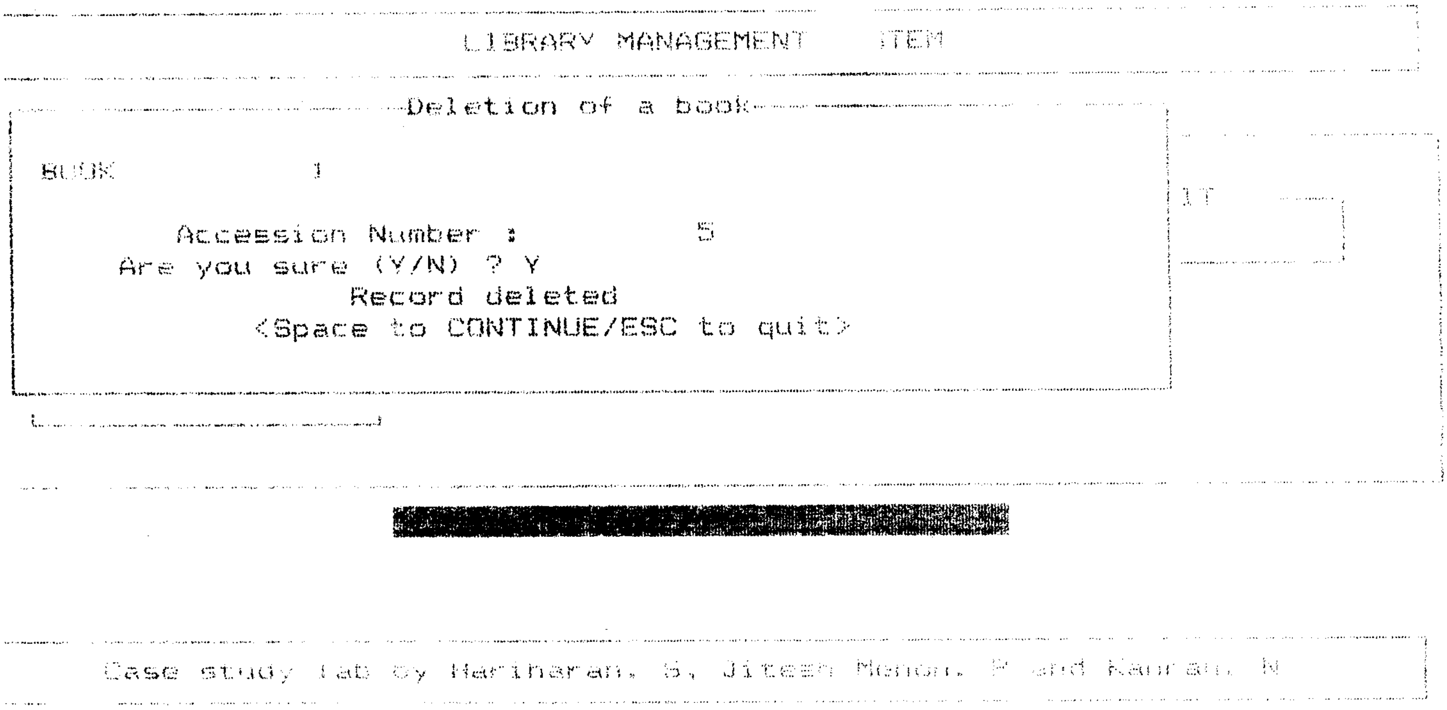
```
                         LIBRARY MANAGEMENT SYSTEM

                          Control Menu for LMS
   BOOK DETAILS       MEMBER DETAILS       HELP          QUIT
  Add a book
  Delete a book
  Search for a book
  Sort book  ----- Search ------
  Display b| Search using Book name
           | Search using Author

      Case study lab by Hariharan. S, Jitesh Menon. P and Kannan. N
```



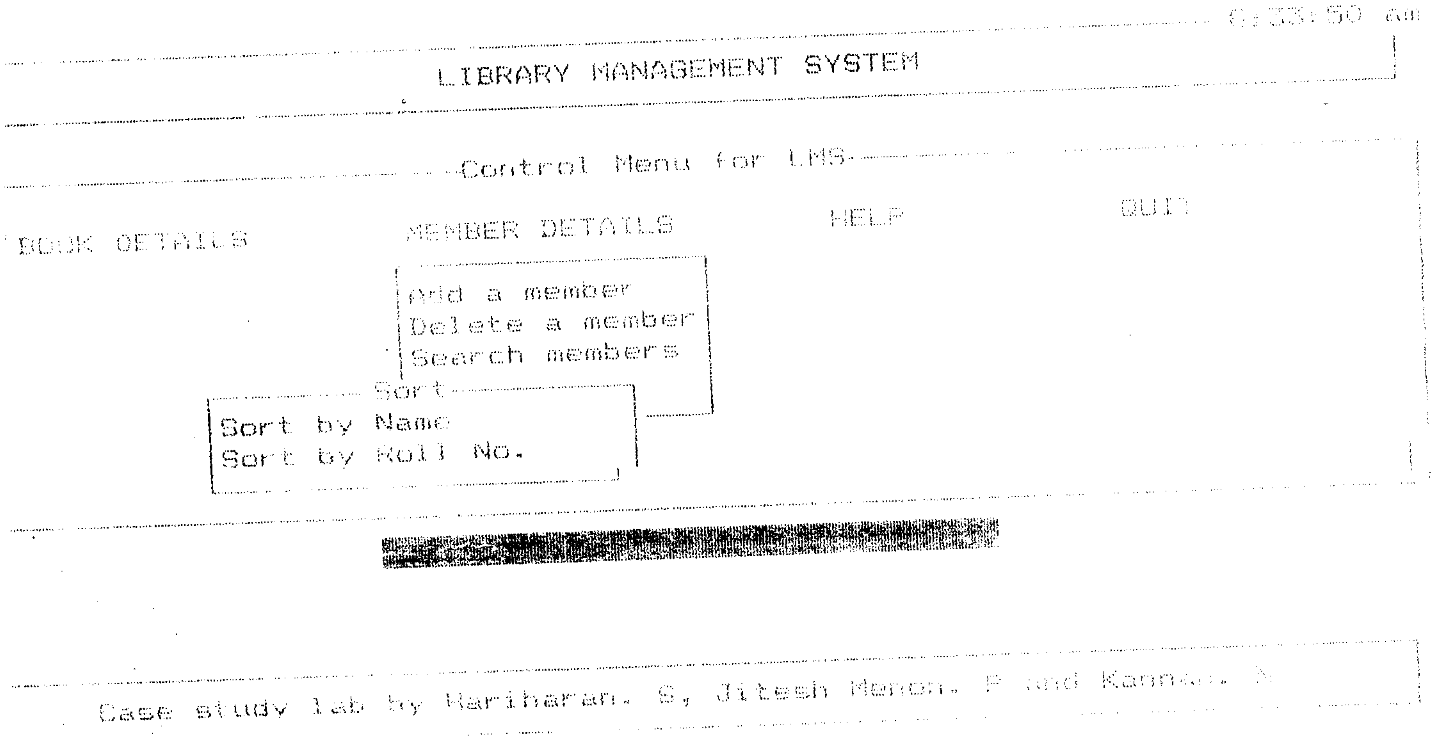
```
                         LIBRARY MANAGEMENT SYSTEM

                          Control Menu for LMS
   BOOK DETAILS       MEMBER DETAILS       HELP          QUIT
  Add a book
  Delete a book
  Search for a book
  Sort book  ----- Sort ------
  Display b| Sort by Book name
           | Sort by Author

      Case study lab by Hariharan. S, Jitesh Menon. P and Kannan. N
```



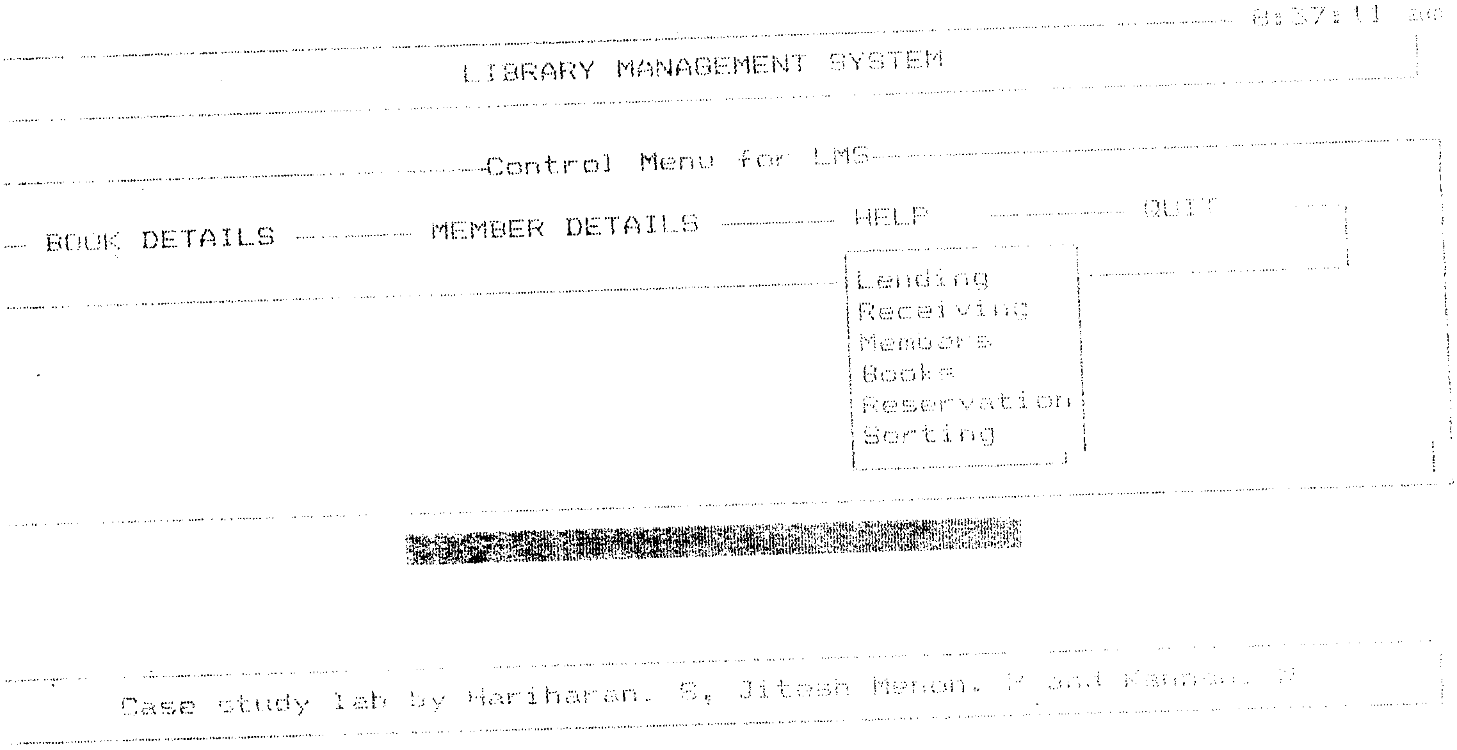
```
                         LIBRARY MANAGEMENT SYSTEM

                         Addition of a book
 information for book          1

  Book Name    :  BORLAND C++
  Author       :  PETERSON
  Volume       :    3
  Edition      :  1995
  Subject      :  COMPUTER
  Copies       :   2
  Text/Ref.    :  T
  Bank/Gen.    :  B
  Call No.     :  34
  Accn. No.    :          9885
  Available    :  Y
  Publisher    :  WAITE GROUP PRESS
  Price        :   150.00
  Date         :  11/11/95
  Agents       :  HIGGIN BOTHAMS
  Bill No.     :  23
```



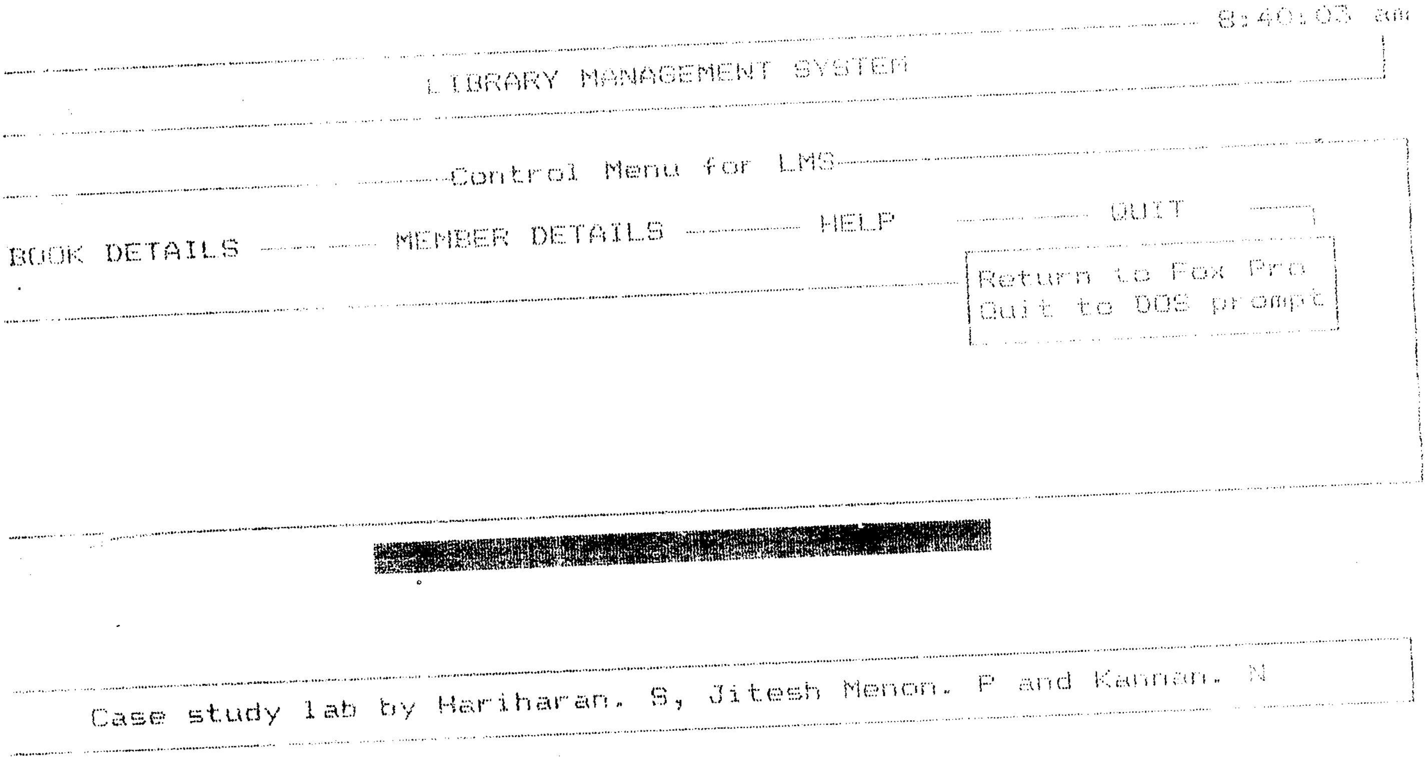
```
                          LIBRARY MANAGEMENT   ITEM

                      Deletion of a book
  BOOK          1

         Accession Number :          5
      Are you sure (Y/N) ? Y
                Record deleted
            <Space to CONTINUE/ESC to quit>

      Case study lab by Hariharan. S, Jiteesh Menon. P and Kannan. N
```



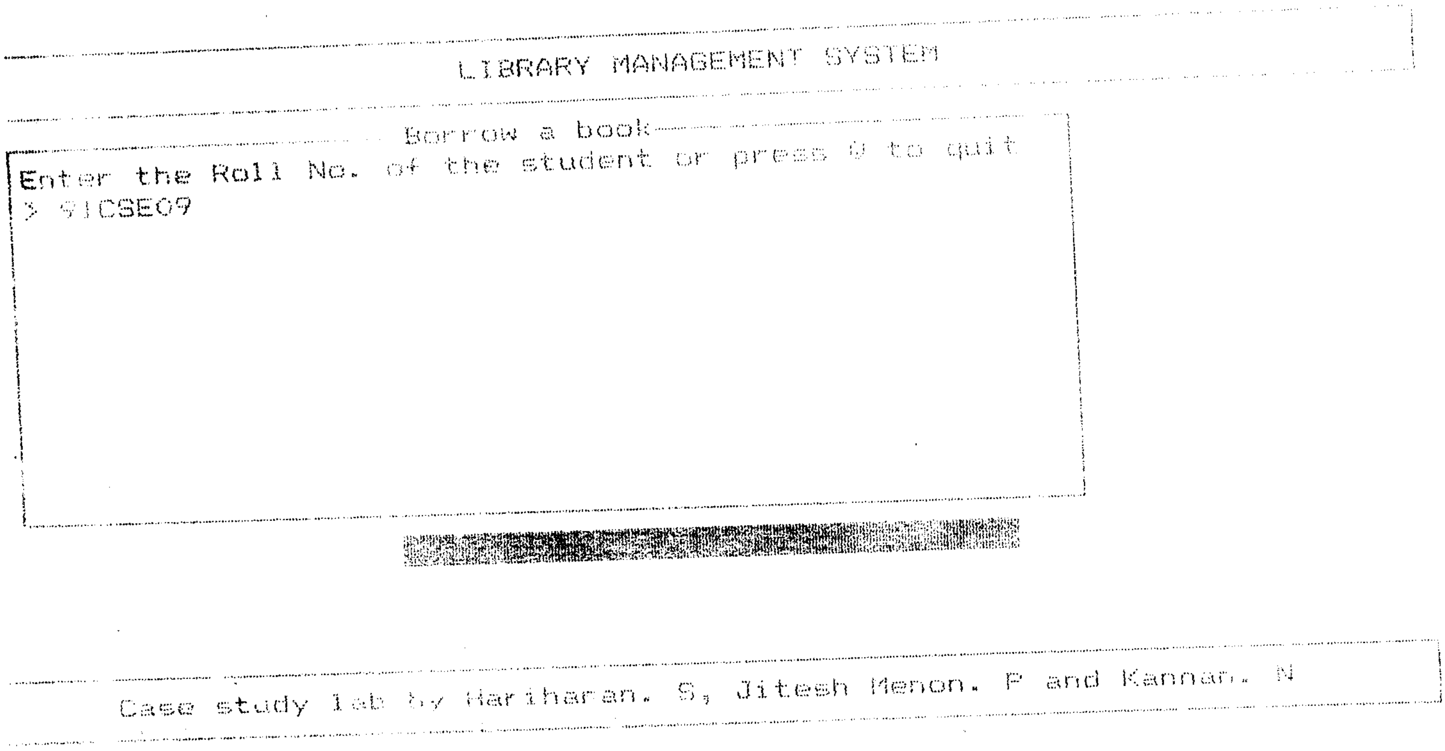
Case study lab by Hariharan. S, Jitesh Menon. P and Kannan. N



Case study lab by Hariharan. S, Jitesh Menon. P and Kannan. N



```
8:40:03 am
LIBRARY MANAGEMENT SYSTEM
Control Menu for LMS

Case study lab by Hariharan. S, Jitesh Menon. P and Kannan. N
```



```
LIBRARY MANAGEMENT SYSTEM
Borrow a book
Enter the Roll No. of the student or press 0 to quit
> 91CSE09

Case study lab by Hariharan. S, Jitesh Menon. P and Kannan. N
```