



1988



# **NEAR EAST UNIVERSITY**

**FACULTY OF ENGINEERING**

**DEPARTMENT OF COMPUTER ENGINEERING**

**COM 400**

**PROJECT TITLE:**

**LIBRARY AUTOMATION**

**SUPERVISOR:**

**MR. OKAN DONANGIL**

**SUBMITTED BY:**

**Name: HAIDAR DOZANDEH**

**ST\_NO: 960588**

**DATE:**

**January 2000**

## **ACKNOWLEDGMENT**

I am greatly indebted to Mr.Okan Donangil, my supervisor, for his generous help throughout the work on this project and the other courses. I will never forget the joyful moments in his lectures on Data Structure.

A very special thanks to my brothers Abdulla, Mohammed Sherif, Amanalla and Abdulaziz for them support in all aspects.

A very special thanks to my teachers Mr.tayseer Al-shanable, prof.Khalil Ismailof, Dr.Adnan Khashman, Mr.Mehrdad Khaledi, Mr.Ozgur Ozderem and my supervisor Mr.okan Donangil for their help during my studies.

I would also like to extend thanks and appreciations to my friends Mohammed, Mousin, Khaled, Samy, Majed, Ramiz, Hisham, Hamdi, Ibaid and Basil for their help during my studies and Imran Sohail for him help in writing this text.

## INTRODUCTION TO LIBRARY

There is one critical word, which can be used to describe the present day word "integration", the introduction of new information technologies has probably been the single most important factor which caused this integration. That is why we are often told that we are living in an "Information Age". It is a well-known fact that the development of advanced of fast speed computer systems has been the main driving force behind the "Information Age".

Therefore, it is not an exaggeration to say that the observed acceleration in the process of integration of world economies and a direct product of "Information Age".

The information has been the result of hard work throughout past ages to achieve this great age. The aspects of the "Information Age" differs and they are too much among which the high speed of interchanging information between countries, invention of the computer, the wide-use of internet which eases like the work of all the members of the society. But still among these aspects the most traditional of useful and the library which is the main source of information for the library to be effective and the active good reputation there must be well-organized of the word and to contain as much books as possible with the recent books of published by the international presses of the world.

For the library to be useful and offers the students desire as long the years he or she is completing his or her studies.

To begin with first of all the library must be equipped with the softy dense that completely avoid the contents of the library form damaging that may be eased due to bumming or any other incidents for this new technologies of offering softy must be available. The library must be automatically and throughout computers connected to the fire-status

In order to put-off any fire accident without any delay in time to protect the contents.

During this time the controlled contains that are supposed to be available in the library and avoid the extending of the fire to the larches of the library.

The library must be divided into branches each branch is responsible about the needs of each department with fully result books.

Now, lets speak about the organize of the library.

### **The general manager:**

The general manger is responsible about the general policy of the library and all branch mangers are related to the ground manager. The general manager plays all the role of the ordinate between the educational managers of the university from one side the international presses of the world from the other side.

The general manager is responsible in an indirect way about the work of the university by receiving reports that are weekly or monthly and upon which decides quickly and correctly according to the situation being faced.

### **Branches managers:**

The branch managers must be appointed according to their practice and experience in the branch they work at.

For example the manager of the electrical engineering department must be have the enough experience and has a well-known of the international process responsible for the



publishing experience and has a well known of the international presses responsible for publishing books related to this field.

This brand is from one side from the other side the branch manger is in direct contact with the workers at his branch and their workers are:

**Controlling compute system workers:**

These workers are responsible about counting the income books, outcome books and equipping the computers with recent programs that enable the student to find his or her need in the shortest possible time and also to provide the web-pages addresses that may help the student in his or her researches.

**General workers:**

There workers are responsible about arranging the books in a well-organized way that makes all the books appear in a very clear way for the easing the work of the student. Others workers who are responsible for the controlling of selling books through which work can be much easier.

**Secretaries:**

Secretaries play an important role in which all the works in other branches are related to them. So, secretaries are responsible for connecting all the branches together in order for all work to be under the control at one system.

The controlling process include the tittles of the books available at the library and to deliver reports of the general manager daily weekly or monthly according to the policy of allowed from one side and from other side to control the process of resulting books in order to put this process always under control.

**Financial manager:**

Responsible about a financial policy of the work in order to achieve the best quality works with the best financial result. The financial manager is responsible about delivering reports to the general manage to reflect the situation and the way that must be followed from a coming period of time.

## INTRODUCTION

The aim of this project was to create a program that could be used for automation at circulation desk of any library. The reason behind this is to make the operation accurate, organized and faster which result in saving time and efforts.

The steps that I have followed in achieving these goals are summarized below:

First of all, I went to the library and met those who are in charge and asked some questions about the system used in the library. Such as, how to add new books and how the system used in searching for the books performed individually book by book on the shelves or is it one of the computer's tasks. By obtaining the answers for these questions I managed to built up an idea about how the operation runs over there, as a result I translated the operation into a software programming.

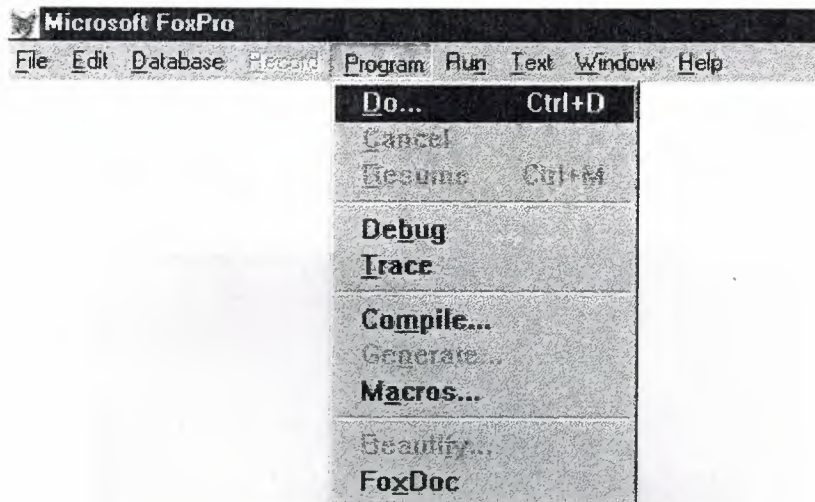
Here FoxPro was selected as the software to create this program, as it is powerful in creating large databases.

The program generally gives the privilege of automation in:

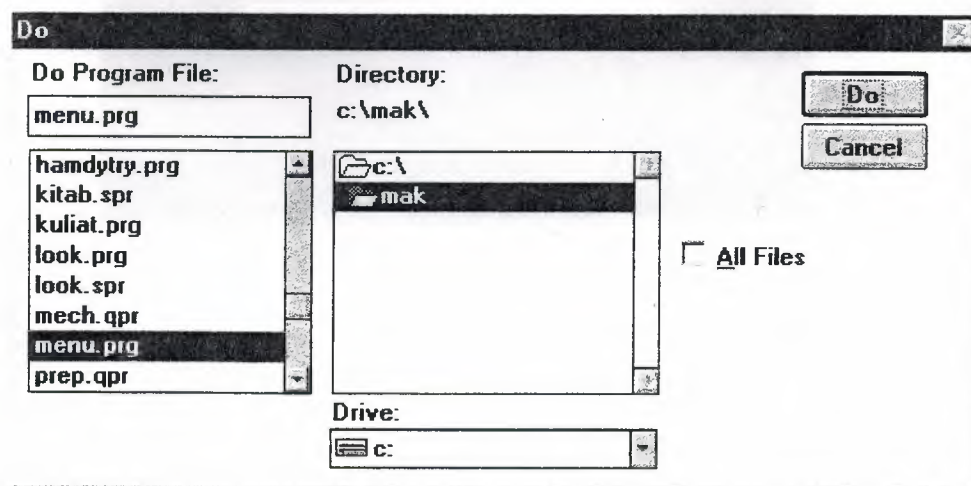
- A) Book Registration
- B) Book search
- C) Member registration and search for it
- D) Subject searching
- E) Borrow processing

## 1. How To Start The Program

Here illustrate how we can use the program. When we are starting the FoxPro for windows we can click the Program Option and do the program.



After the above step we can find the directory of our program and find the *menu.prg* and Do the program and we can use our general program as shown below:

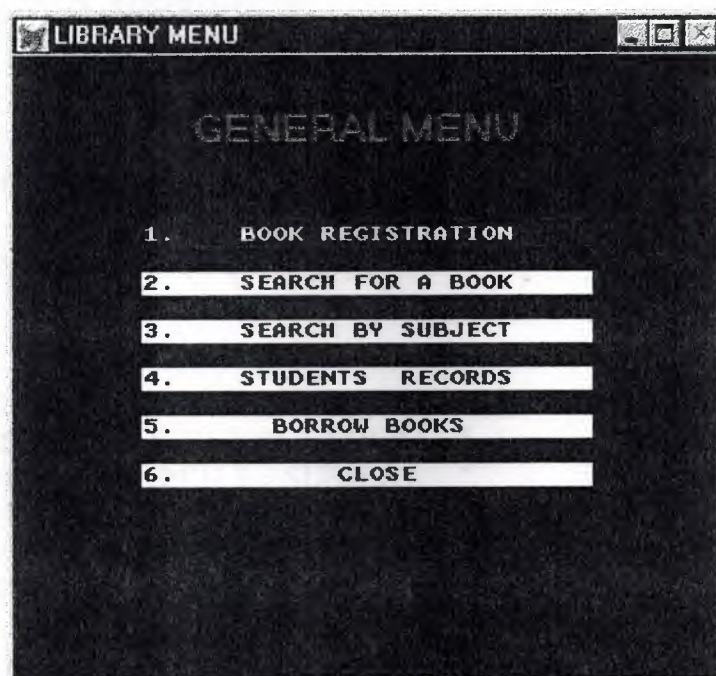




## 2. Main Menu

The aim of the main menu is to use the program easily, faster and use all the process screens or necessary programs in the same time. In order to use this menu, from the program menu select the DO option and choose the *menu.prg* and run the program (1. How To Start The Program). After running the program we will see the menu as 2. *Main Menu*.

Now the menu is ready to be use, just press the icon you want and the screen of your choice will be shown. When we want to exit the program, we click the *close* button.



## Program how to create the main menu

\*CLEAR

SET SYSMENU OFF

SET SAFETY OFF

SET TALK OFF

SET STATUS BAR OFF

SET SYSMENU ON

DEFINE WINDOW system FROM 3,15 TO 30,60 SYSTEM;  
FLOAT SHADOW TITLE "LIBRARY MENU" IN SCREEN;  
GROW MINIMIZE ZOOM CLOSE COLOR R++  
ACTIVATE WINDOW system

SET TALK OFF

STORE 0 TO CHOICE

DO WHILE CHOICE <= 5

@2, 6 SAY " GENERAL MENU" ;

FONT "MS Sans Serif", 14 ;

STYLE "BT+" ;

COLOR RGB(0,255,128,,,) )

@7, 8 PROMPT "1. BOOK REGISTRATION "

@9, 8 PROMPT "2. SEARCH FOR A BOOK "

@11, 8 PROMPT "3. SEARCH BY SUBJECT "

@13, 8 PROMPT "4. STUDENTS RECORDS "

@15, 8 PROMPT "5. BORROW BOOKS "

@17, 8 PROMPT "6. CLOSE "

MENU TO CHOICE

DO CASE

CASE CHOICE=1

DO C:\mak\kitab.spr

CASE CHOICE=2

DO C:\mak\hamdytry.prg

CASE CHOICE=3

DO C:\mak\look.spr

CASE CHOICE=4

DO C:\mak\talib.spr

CASE CHOICE=5

DO C:\mak\dain.spr

CASE CHOICE=6

DEACTIVATE WINDOW system

RELEASE WINDOW system

ENDCASE

ENDDO

\*CLEAR



### 3. BOOKS REGISTRATION

Book Registration menu is used for *Adding* the new books or *Deleteing* the damaged ones and containing the information about the books like (Book Name, Serial number, Author, no of copy and so on which is shown in the screen). Book Registration screen is shown as 3. Book Registration:

We can use Book Registration screen directly from *Program menu*, *DO*, *Kitab.spr* and run the screen, or we can use it from the *main menu* by clicking the Book Registration button and get it very easily.

The screenshot shows a graphical user interface for book registration. At the top, there is a window titled 'LIBRARY MENU' and a sub-window titled 'GENERAL MENU'. The main window is titled 'BOOKS REGISTRATION'. It contains a form with the following fields and values:

Field	Value
Book Name:	BUILDING STRUCTURE
Serial No:	6
Author Name:	ALI
Book Type:	CIVIL
Subject:	INFORMATION
Isbn:	56-6958-5K
Edition:	4TH
Publishing Year:	1995
Publisher:	NEU
No Of Copies:	3
Avilable:	1

At the bottom of the form, there is a row of buttons: Top, Prev, Next, End, Locate, Add, Edit, Delete, Print, and Close.

## Program how to create the books registration

```

* *****
*
* * 01/21/00          KITAB.SPR          21:43:53
*
* *****
*
* * Author's Name
*
* * Copyright (c) 2000 Company Name
* * Address
* * City,   Zip
*
* * Description:
* * This program was automatically generated by GENSCRN.
*
* *****

* *****
*
* *          KITAB/Windows Setup Code - SECTION 1
*
* *****
*

```

```

#REGION 1
PRIVATE wzfields,wztalk
IF SET("TALK") = "ON"
    SET TALK OFF
    m.wztalk = "ON"
ELSE
    m.wztalk = "OFF"
ENDIF
m.wzfields=SET('FIELDS')
SET FIELDS OFF
IF m.wztalk = "ON"
    SET TALK ON
ENDIF

```

```

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"

```

```

        SET TALK OFF
        m.talkstat = "ON"
ELSE
        m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.rborder = SET("READBORDER")
SET READBORDER ON

m.currarea = SELECT()

*      *****
*      *
*      *      S8234433/Windows Databases, Indexes, Relations
*      *
*      *****
*

IF USED("kitab")
        SELECT kitab
        SET ORDER TO TAG "bn"
ELSE
        SELECT 0
        USE (LOCFILE("kitab.dbf","DBF","Where is kitab?"));
        AGAIN ALIAS kitab ;
        ORDER TAG "bn"
ENDIF

*      *****
*      *
*      *      Windows Window definitions
*      *
*      *****
*

IF NOT WEXIST("_0011aku2v")
        DEFINE WINDOW _0011aku2v ;
                AT 0.000, 0.000 ;
                SIZE 26.077, 83.333 ;
                TITLE "" ;
                FONT "MS Sans Serif", 8 ;
                STYLE "B" ;

```



```

        FLOAT ;
        CLOSE ;
        MINIMIZE ;
        COLOR RGB(,,,192,192,192)
    MOVE WINDOW _0011aku2v CENTER
ENDIF

```

```

* *****
*
* *      KITAB/Windows Setup Code - SECTION 2
* *
* *****
*

```

```
#REGION 1
```

```

#DEFINE C_DBFEMPTY      'Database is empty, add a record?'
#DEFINE C_EDITS          'Please finish your edits.'
#DEFINE C_TOPFILE        'Top of file.'
#DEFINE C_ENDFILE        'End of file.'
#DEFINE C_BRTITLE        'Locate Record'
#DEFINE C_NOLOCK         'Sorry, could not lock record -- try again later.'
#DEFINE C_ECANCEL        'Edits Canceled.'
#DEFINE C_DELREC         'Delete selected record?'
#DEFINE C_NOFEAT         'Feature not available yet.'
#DEFINE C_NOWIZ          'Wizard application is not available.'
#DEFINE C_MAKEREPO       'Creating report with Report Wizard.'
#DEFINE C_NOREPO         'Could not create report.'
#DEFINE C_DELNOTE        'Deleting records...'
#DEFINE C_READONLY       'Table is read-only. No editing allowed.'
#DEFINE C_NOTABLE        'No table selected. Open table or run query.'
#DEFINE C_BADEXP         'Invalid expression.'
#DEFINE C_LOCWIZ         'Locate WIZARD.APP:'
#DEFINE C_MULTITABLE     'You have multiple related tables. Adding records in not
allowed.'

```

```

MOVE WINDOW ' _0011aku2v' CENTER
PRIVATE isediting,isadding,wztblarr
PRIVATE wzolddelete,wzolderror,wzoldesc
PRIVATE wzalias, tempcurs,wzlastrec
PRIVATE isreadonly,find__drop,is2table

```

```

IF EMPTY(ALIAS())
    WAIT WINDOW C__NOTABLE
    RETURN

```

ENDIF

```
m.wztblarr= "  
m.wzalias=SELECT()  
m.isediting=.F.  
m.isadding=.F.  
m.is2table = .F.  
m.wzolddelete=SET('DELETE')  
SET DELETED ON  
m.tempcurs=SYS(2015) &&used if General field  
m.wzlastrec = 1  
m.wzolderror=ON('error')  
ON ERROR DO wizerrorhandler  
wzoldesc=ON('KEY','ESCAPE')  
ON KEY LABEL ESCAPE  
m.find_drop = IIF(_DOS,0,2)
```

```
m.isreadonly=IIF(ISREAD(),.T.,.F.)  
IF m.isreadonly  
    WAIT WINDOW C_READONLY TIMEOUT 1  
ENDIF
```

```
IF RECCOUNT()=0 AND !m.isreadonly AND fox_alert(C_DBFEMPTY)  
    APPEND BLANK  
ENDIF
```

```
GOTO TOP  
SCATTER MEMVAR MEMO
```

```
* *****  
* *  
* *          KITAB/Windows Screen Layout  
* *  
* *****  
*
```

```
#REGION 1  
IF WVISIBLE("_0011aku2v")  
    ACTIVATE WINDOW _0011aku2v SAME  
ELSE  
    ACTIVATE WINDOW _0011aku2v NOSHOW  
ENDIF  
@ 2.692,0.000 TO 2.692,83.333 ;  
    PEN 2, 8 ;  
    STYLE "1"
```

@ 22.231,0.000 TO 22.231,83.333 ;  
 PEN 2, 8 ;  
 STYLE "1"  
 @ 0.538,22.500 SAY "BOOKS REGISTRATION" ;  
 FONT "times new roman", 14 ;  
 STYLE "BT"  
 @ 3.462,8.333 SAY "Book Name:" ;  
 SIZE 0.938,18.000 ;  
 FONT "Times New Roman", 9 ;  
 STYLE "BT" ;  
 PICTURE "@J" ;  
 COLOR RGB(,,255,255,255)  
 @ 3.538,26.333 GET m.book\_name ;  
 SIZE 1.000,42.000 ;  
 DEFAULT " " ;  
 FONT "Times New Roman", 8 ;  
 STYLE "B" ;  
 PICTURE "@K XX" ;  
 WHEN isediting ;  
 COLOR ,RGB(0,0,0,255,255,255)  
 @ 5.231,8.333 SAY "Serial No:" ;  
 SIZE 0.938,18.000 ;  
 FONT "Times New Roman", 9 ;  
 STYLE "BT" ;  
 PICTURE "@J" ;  
 COLOR RGB(,,255,255,255)  
 @ 5.308,26.333 GET m.serial\_no ;  
 SIZE 1.000,12.000 ;  
 DEFAULT " " ;  
 FONT "Times New Roman", 8 ;  
 STYLE "B" ;  
 PICTURE "@K 9999999999" ;  
 WHEN isediting ;  
 COLOR ,RGB(0,0,0,255,255,255)  
 @ 6.923,8.333 SAY "Author Name:" ;  
 SIZE 0.938,18.000 ;  
 FONT "Times New Roman", 9 ;  
 STYLE "BT" ;  
 PICTURE "@J" ;  
 COLOR RGB(,,255,255,255)  
 @ 7.000,26.333 GET m.author ;  
 SIZE 1.000,42.000 ;  
 DEFAULT " " ;  
 FONT "Times New Roman", 8 ;  
 STYLE "B" ;  
 PICTURE "@K XX" ;



```

        WHEN isediting ;
        COLOR ,RGB(0,0,0,255,255,255)
@ 8.615,8.333 SAY "Book Type:" ;
        SIZE 0.938,18.000 ;
        FONT "Times New Roman", 9 ;
        STYLE "BT" ;
        PICTURE "@J" ;
        COLOR RGB(,,,255,255,255)
@ 8.692,26.333 GET m.book_type ;
        SIZE 1.000,16.000 ;
        DEFAULT " " ;
        FONT "Times New Roman", 8 ;
        STYLE "B" ;
        PICTURE "@K XXXXXXXXXXXX" ;
        WHEN isediting ;
        COLOR ,RGB(0,0,0,255,255,255)
@ 10.154,8.333 SAY "Subject:" ;
        SIZE 0.938,18.000 ;
        FONT "Times New Roman", 9 ;
        STYLE "BT" ;
        PICTURE "@J" ;
        COLOR RGB(,,,255,255,255)
@ 10.231,26.333 GET m.subject ;
        SIZE 1.000,42.000 ;
        DEFAULT " " ;
        FONT "Times New Roman", 8 ;
        STYLE "B" ;
        PICTURE "@K XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" ;
        WHEN isediting ;
        COLOR ,RGB(0,0,0,255,255,255)
@ 11.769,8.333 SAY "Isbn:" ;
        SIZE 0.938,18.000 ;
        FONT "Times New Roman", 9 ;
        STYLE "BT" ;
        PICTURE "@J" ;
        COLOR RGB(,,,255,255,255)
@ 11.846,26.333 GET m.isbn ;
        SIZE 1.000,16.000 ;
        DEFAULT " " ;
        FONT "Times New Roman", 8 ;
        STYLE "B" ;
        PICTURE "@K XXXXXXXXXXXX" ;
        WHEN isediting ;
        COLOR ,RGB(0,0,0,255,255,255)
@ 13.538,8.333 SAY "Edition:" ;
        SIZE 0.938,18.000 ;

```

```

FONT "Times New Roman", 9 ;
STYLE "BT" ;
PICTURE "@J" ;
COLOR RGB(,,255,255,255)
@ 13.615,26.333 GET m.edition ;
SIZE 1.000,16.000 ;
DEFAULT " " ;
FONT "Times New Roman", 8 ;
STYLE "B" ;
PICTURE "@K XXXXXXXXXXXXX" ;
WHEN isediting ;
COLOR ,RGB(0,0,0,255,255,255)
@ 15.231,8.333 SAY "Publishing Year:" ;
SIZE 0.938,18.000 ;
FONT "Times New Roman", 9 ;
STYLE "BT" ;
PICTURE "@J" ;
COLOR RGB(,,255,255,255)
@ 15.308,26.333 GET m.pub_year ;
SIZE 1.000,5.000 ;
DEFAULT " " ;
FONT "Times New Roman", 8 ;
STYLE "B" ;
PICTURE "@K 9999" ;
WHEN isediting ;
COLOR ,RGB(0,0,0,255,255,255)
@ 17.000,8.333 SAY "Publisher:" ;
SIZE 0.938,18.000 ;
FONT "Times New Roman", 9 ;
STYLE "BT" ;
PICTURE "@J" ;
COLOR RGB(,,255,255,255)
@ 17.077,26.333 GET m.publisher ;
SIZE 1.000,21.000 ;
DEFAULT " " ;
FONT "Times New Roman", 8 ;
STYLE "B" ;
PICTURE "@K XXXXXXXXXXXXXXXXXXXX" ;
WHEN isediting ;
COLOR ,RGB(0,0,0,255,255,255)
@ 18.692,8.333 SAY "No Of Copies:" ;
SIZE 0.938,18.000 ;
FONT "Times New Roman", 9 ;
STYLE "BT" ;
PICTURE "@J" ;
COLOR RGB(,,255,255,255)

```

```

@ 18.769,26.333 GET m.no_of_copy ;
    SIZE 1.000,15.600 ;
    DEFAULT " " ;
    FONT "Times New Roman", 8 ;
    STYLE "B" ;
    PICTURE "@K 9999999999" ;
    WHEN isediting ;
    COLOR ,RGB(0,0,0,255,255,255)
@ 20.462,8.333 SAY "Avilable:" ;
    SIZE 0.938,18.000 ;
    FONT "Times New Roman", 9 ;
    STYLE "BT" ;
    PICTURE "@J" ;
    COLOR RGB(,,255,255,255)
@ 20.538,26.333 GET m.avilable ;
    SIZE 1.000,15.600 ;
    DEFAULT " " ;
    FONT "Times New Roman", 8 ;
    STYLE "B" ;
    PICTURE "@K 9999999999" ;
    WHEN ROSY() ;
    COLOR ,RGB(0,0,0,255,255,255)
@ 23.385,1.667 GET m.top_btn ;
    PICTURE "@*HN \<Top" ;
    SIZE 1.769,7.833,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    VALID btn_val('TOP') ;
    MESSAGE 'Go to first record.'
@ 23.385,9.667 GET m.prev_btn ;
    PICTURE "@*HN \<Prev" ;
    SIZE 1.769,7.833,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    VALID btn_val('PREV') ;
    MESSAGE 'Go to previous record.'
@ 23.385,17.667 GET m.next_btn ;
    PICTURE "@*HN \<Next" ;
    SIZE 1.769,7.833,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    VALID btn_val('NEXT') ;
    MESSAGE 'Go to next record.'

```



```

@ 23.385,25.667 GET m.end_btn ;
    PICTURE "@*HN \<End" ;
    SIZE 1.769,7.833,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    VALID btn_val('END') ;
    MESSAGE 'Go to last record.'
@ 23.385,33.667 GET m.loc_btn ;
    PICTURE "@*HN \<Locate" ;
    SIZE 1.769,7.833,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    VALID btn_val('LOCATE') ;
    MESSAGE 'Locate a record.'
@ 23.385,41.667 GET m.add_btn ;
    PICTURE "@*HN \<Add" ;
    SIZE 1.769,7.833,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    VALID btn_val('ADD') ;
    MESSAGE 'Add a new record.'
@ 23.385,49.667 GET m.edit_btn ;
    PICTURE "@*HN Ed\<it" ;
    SIZE 1.769,7.833,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    VALID btn_val('EDIT') ;
    MESSAGE 'Edit current record.'
@ 23.385,57.667 GET m.del_btn ;
    PICTURE "@*HN \<Delete" ;
    SIZE 1.769,7.833,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    VALID btn_val('DELETE') ;
    MESSAGE 'Delete current record.'
@ 23.385,65.667 GET m.prnt_btn ;
    PICTURE "@*HN P\<rint" ;
    SIZE 1.769,7.833,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;

```

```

        VALID btn_val('PRINT') ;
        MESSAGE 'Print report.'
@ 23.385,73.667 GET m.exit_btn ;
        PICTURE "@*HN \<Close" ;
        SIZE 1.769,7.833,0.667 ;
        DEFAULT 1 ;
        FONT "MS Sans Serif", 8 ;
        STYLE "B" ;
        VALID btn_val('EXIT') ;
        MESSAGE 'Close screen.'

```

```

IF NOT WVISIBLE("_00l1aku2v")
        ACTIVATE WINDOW _00l1aku2v
ENDIF

```

```

* *****
*
* *   WindowsREAD contains clauses from SCREEN s8234433
* *
* *****
*

```

```

READ CYCLE ;
        ACTIVATE READACT() ;
        DEACTIVATE READDEAC() ;
        NOLOCK

```

```

RELEASE WINDOW _00l1aku2v

```

```

* *****
*
* *           Windows Closing Databases
* *
* *****
*

```

```

IF USED("kitab")
        SELECT kitab
        USE
ENDIF

```

```

SELECT (m.currarea)

```

```

#REGION 0

```

```
SET READBORDER &rborder
```

```
IF m.talkstat = "ON"
```

```
    SET TALK ON
```

```
ENDIF
```

```
IF m.compstat = "ON"
```

```
    SET COMPATIBLE ON
```

```
ENDIF
```

```
* *****
*
*      KITAB/Windows Cleanup Code
*
* *****
*
```

```
#REGION 1
```

```
SET DELETED &wzolddelete
```

```
SET FIELDS &wzfields
```

```
ON ERROR &wzolderror
```

```
ON KEY LABEL ESCAPE &wzoldesc
```

```
DO CASE
```

```
CASE _DOS AND SET('DISPLAY')='VGA25'
```

```
    @24,0 CLEAR TO 24,79
```

```
CASE _DOS AND SET('DISPLAY')='VGA50'
```

```
    @49,0 CLEAR TO 49,79
```

```
CASE _DOS
```

```
    @24,0 CLEAR TO 24,79
```

```
ENDCASE
```

```
****Procedures****
```

```
* *****
*
*      KITAB/Windows Supporting Procedures and Functions
*
* *****
*
```

```
#REGION 1
```

```
PROCEDURE readdeac
```

```
    IF isediting
```

```
        ACTIVATE WINDOW '_00l1aku2v'
```



```

    WAIT WINDOW C_EDITS NOWAIT
ENDIF
IF !WVISIBLE(WOUTPUT())
    CLEAR READ
    RETURN .T.
ENDIF
RETURN .F.

```

```

PROCEDURE readact
    IF !isediting
        SELECT (m.wzalias)
        SHOW GETS
    ENDIF
    DO REFRESH
RETURN

```

```

PROCEDURE wizerrorhandler
    WAIT WINDOW message()
RETURN

```

```

PROCEDURE printrec
    PRIVATE sOldError,wizfname,saverec,savearea,tmpcurs,tmpstr
    PRIVATE prnt_btn,p_recs,p_output,pr_out,pr_record
    STORE 1 TO p_recs,p_output
    STORE 0 TO prnt_btn
    STORE RECNO() TO saverec
    m.sOldError=ON('error')
    DO pdialog
    IF m.prnt_btn = 2
        RETURN
    ENDIF
    IF !FILE(ALIAS()+'FRX')
        m.wizfname=SYS(2004)+'WIZARDS\'+'WIZARD.APP'
        IF !FILE(m.wizfname)
            ON ERROR *
            m.wizfname=LOCFILE('WIZARD.APP','APP',C_LOCWIZ)
            ON ERROR &sOldError
            IF !'WIZARD.APP'$UPPER(m.wizfname)
                WAIT WINDOW C_NOWIZ
                RETURN
            ENDIF
        ENDIF
    ENDIF
    WAIT WINDOW C_MAKEREPO NOWAIT
    m.savearea=SELECT()
    m.tmpcurs='_'+LEFT(SYS(3),7)

```

```

        CREATE CURSOR (m.tmpcurs) (comment m)
        m.tmpstr = '* LAYOUT = COLUMNAR'+CHR(13)+CHR(10)
        INSERT INTO (m.tmpcurs) VALUES(m.tmpstr)
        SELECT (m.savearea)
        DO (m.wizfname) WITH
        ", 'WZ_QREPO', 'NOSCRN/CREATE', ALIAS(), m.tmpcurs
        USE IN (m.tmpcurs)
    WAIT CLEAR
        IF !FILE(ALIAS()+'.FRX') && wizard could not create report
        WAIT WINDOW C_NOREPO
        RETURN
        ENDIF
    ENDIF

    m.pr_out=IIF(m.p_output=1, 'TO PRINT NOCONSOLE', 'PREVIEW')
    m.pr_record=IIF(m.p_recs=1, 'NEXT 1', 'ALL')
    REPORT FORM (ALIAS()) &pr_out &pr_record
    GO m.saverec
RETURN

```

```

PROCEDURE BTN_VAL
    PARAMETER m.btnname
    DO CASE
    CASE m.btnname='TOP'
        GO TOP
        WAIT WINDOW C_TOPFILE NOWAIT
    CASE m.btnname='PREV'
        IF !BOF()
            SKIP -1
        ENDIF
        IF BOF()
            WAIT WINDOW C_TOPFILE NOWAIT
            GO TOP
        ENDIF
    CASE m.btnname='NEXT'
        IF !EOF()
            SKIP 1
        ENDIF
        IF EOF()
            WAIT WINDOW C_ENDFILE NOWAIT
            GO BOTTOM
        ENDIF
    CASE m.btnname='END'
        GO BOTTOM
        WAIT WINDOW C_ENDFILE NOWAIT
    
```

```

CASE m.btnname='LOCATE'
DO loc_dlog
CASE m.btnname='ADD' AND !isediting &&add record
isediting=.T.
isadding=.T.
=edithand('ADD')
_curobj=1
DO refresh
SHOW GETS
RETURN
CASE m.btnname='EDIT' AND !isediting &&edit record
IF EOF() OR BOF()
WAIT WINDOW C_ENDFILE NOWAIT
RETURN
ENDIF
IF RLOCK()
isediting=.T.
_curobj=1
DO refresh
RETURN
ELSE
WAIT WINDOW C_NOLOCK
ENDIF
CASE m.btnname='EDIT' AND isediting &&save record
IF isadding
=edithand('SAVE')
ELSE
GATHER MEMVAR MEMO
ENDIF
UNLOCK
isediting=.F.
isadding=.F.
DO refresh
CASE m.btnname='DELETE' AND isediting &&cancel record
IF isadding
=edithand('CANCEL')
ENDIF
isediting=.F.
isadding=.F.
UNLOCK
WAIT WINDOW C_ECANCEL NOWAIT
DO refresh
CASE m.btnname='DELETE'
IF EOF() OR BOF()
WAIT WINDOW C_ENDFILE NOWAIT
RETURN

```



```

ENDIF
IF fox_alert(C_DELREC)
    DELETE
    IF !EOF() AND DELETED()
        SKIP 1
    ENDIF
    IF EOF()
        WAIT WINDOW C_ENDFILE NOWAIT
        GO BOTTOM
    ENDIF
ENDIF
CASE m.btnname='PRINT'
    DO printrec
    RETURN
CASE m.btnname='EXIT'
    m.bailout=.T. &&this is needed if used with FoxApp
    CLEAR READ
    RETURN
ENDCASE
SCATTER MEMVAR MEMO
SHOW GETS
RETURN

```

#### PROCEDURE REFRESH

```

DO CASE
CASE m.isreadonly AND RECCOUNT()=0
    SHOW GETS DISABLE
    SHOW GET exit_btn ENABLE
CASE m.isreadonly
    SHOW GET add_btn DISABLE
    SHOW GET del_btn DISABLE
    SHOW GET edit_btn DISABLE
CASE (RECCOUNT()=0 OR EOF()) AND !m.isediting
    SHOW GETS DISABLE
    SHOW GET add_btn ENABLE
    SHOW GET exit_btn ENABLE
CASE m.isediting
    SHOW GET find_drop DISABLE
    SHOW GET top_btn DISABLE
    SHOW GET prev_btn DISABLE
    SHOW GET loc_btn DISABLE
    SHOW GET next_btn DISABLE
    SHOW GET end_btn DISABLE
    SHOW GET add_btn DISABLE
    SHOW GET prnt_btn DISABLE

```

```

    SHOW GET exit_btn DISABLE
    SHOW GET edit_btn,1 PROMPT "\<Save"
    SHOW GET del_btn,1 PROMPT "\<Cancel"
    ON KEY LABEL ESCAPE DO BTN_VAL WITH 'DELETE'
    RETURN
OTHERWISE
    SHOW GET edit_btn,1 PROMPT "Ed\<it"
    SHOW GET del_btn,1 PROMPT "\<Delete"
    SHOW GETS ENABLE
ENDCASE
IF m.is2table
    SHOW GET add_btn DISABLE
ENDIF
ON KEY LABEL ESCAPE
RETURN

PROCEDURE edithand
    PARAMETER m.paction
    DO CASE
    CASE m.paction = 'ADD'
        SCATTER MEMVAR MEMO BLANK
    CASE m.paction = 'SAVE'
        INSERT INTO (ALIAS()) FROM MEMVAR
    CASE m.paction = 'CANCEL'
        * nothing here
    ENDCASE
RETURN

PROCEDURE fox_alert
    PARAMETER wzalrtmess
    PRIVATE alrtbtn
    m.alrtbtn=2
    DEFINE WINDOW _qec1ij2t7 AT 0,0 SIZE 8,50 ;
    FONT "MS Sans Serif",10 STYLE 'B' ;
    FLOAT NOCLOSE NOMINIMIZE DOUBLE TITLE WTITLE()
    MOVE WINDOW _qec1ij2t7 CENTER
    ACTIVATE WINDOW _qec1ij2t7 NOSHOWN
    @ 2,(50-txtwidth(wzalrtmess))/2 SAY wzalrtmess;
    FONT "MS Sans Serif", 10 STYLE "B"
    @ 6,18 GET m.alrtbtn ;
    PICTURE "@*HT \<OK;\?\\<Cancel" ;
    SIZE 1.769,8.667,1.333 ;
    FONT "MS Sans Serif", 8 STYLE "B"
    ACTIVATE WINDOW _qec1ij2t7
    READ CYCLE MODAL

```

```
RELEASE WINDOW _qec1ij2t7
RETURN m.alrtbtn=1
```

```
PROCEDURE pdialog
  DEFINE WINDOW _qjn12zbvh ;
    AT 0.000, 0.000 ;
    SIZE 13.231, 54.800 ;
    TITLE "Microsoft FoxPro" ;
    FONT "MS Sans Serif", 8 ;
    FLOAT NOCLOSE MINIMIZE SYSTEM
  MOVE WINDOW _qjn12zbvh CENTER
  ACTIVATE WINDOW _qjn12zbvh NOSHOW
  @ 2.846, 33.600 SAY "Output:" ;
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
  @ 2.846, 4.800 SAY "Print:" ;
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
  @ 4.692, 7.200 GET m.p_recs ;
    PICTURE "@*RVN \<Current Record;\<All Records" ;
    SIZE 1.308, 18.500, 0.308 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
  @ 4.692, 36.000 GET m.p_output ;
    PICTURE "@*RVN \<Printer;Pre\<view" ;
    SIZE 1.308, 12.000, 0.308 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
  @ 10.154, 16.600 GET m.prnt_btn ;
    PICTURE "@*HT P\<rint;Ca\<ncel" ;
    SIZE 1.769, 8.667, 0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B"
  ACTIVATE WINDOW _qjn12zbvh
  READ CYCLE MODAL
  RELEASE WINDOW _qjn12zbvh
RETURN
```

```
PROCEDURE loc_dlog
  PRIVATE gfields,i
  DEFINE WINDOW wzlocate FROM 1,1 TO 20,40;
```



```

        SYSTEM GROW CLOSE ZOOM FLOAT FONT "MS Sans Serif",8
MOVE WINDOW wzlocate CENTER
m.gfields=SET('FIELDS',2)
IF !EMPTY(RELATION(1))
    SET FIELDS ON
    IF m.gfields # 'GLOBAL'
        SET FIELDS GLOBAL
    ENDIF
    IF EMPTY(FLDLIST())
        m.i=1
        DO WHILE !EMPTY(OBJVAR(m.i))
            IF ATC('M.',OBJVAR(m.i))=0
                SET FIELDS TO (OBJVAR(m.i))
            ENDIF
            m.i = m.i + 1
        ENDDO
    ENDIF
ENDIF
BROWSE WINDOW wzlocate NOEDIT NODELETE ;
    NOMENU TITLE C_BRTITLE
SET FIELDS &gfields
SET FIELDS OFF
RELEASE WINDOW wzlocate
RETURN

```

```

FUNCTION ROSY
CLOSE DATABASES
USE (LOCFILE("dain.dbf","dbf","WHERE IS dain ?"));
    SHARE;
    AGAIN ALIAS dain ;
    ORDER TAG "sno"
SEEK m.serial_no
IF FOUND()
m.avilable=avilable
ENDIF
show gets
CLOSE DATABASES
USE (LOCFILE("kitab.dbf","DBF","Where is kitab?"));
share ;
AGAIN ALIAS kitab ;
ORDER TAG "so"

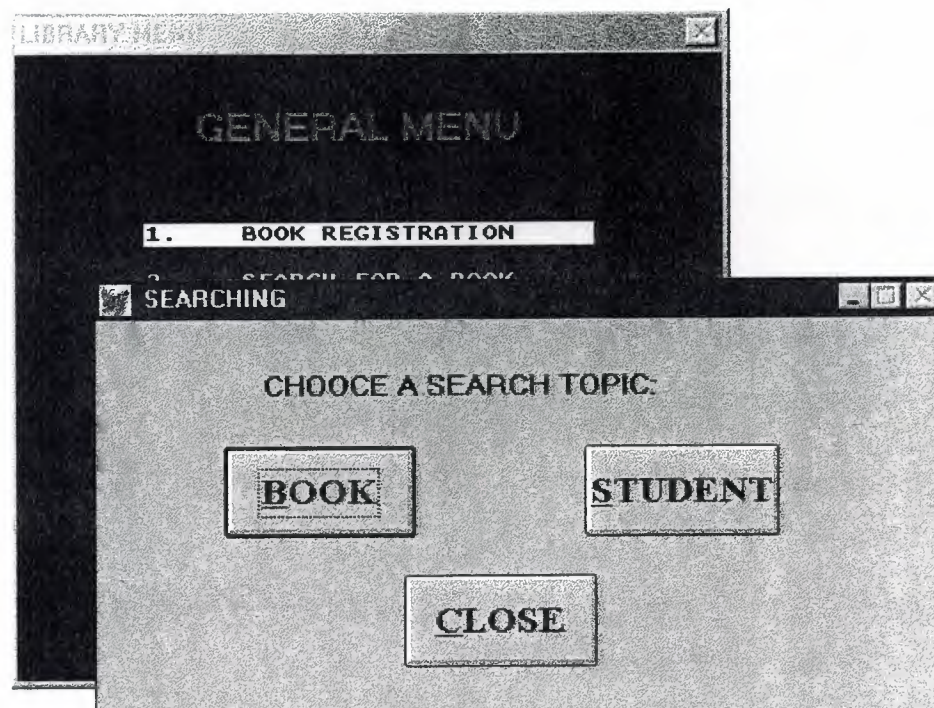
```

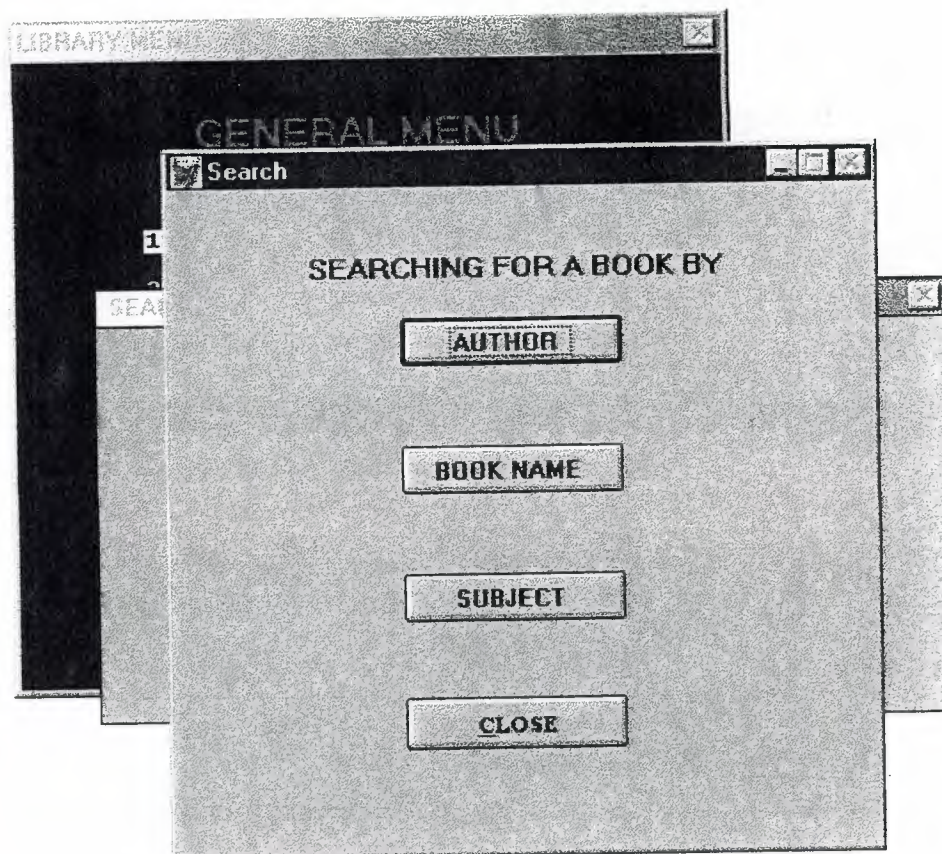
## 4. Searching For A Book

Here we used some methods for searching for the books by Author, Book Name and Subject. This screen uses to find the specific books which he/she wants to borrow it and with all information of that book. This screen uses the *Kitab.dbf* in order to search for the book. In order to search by subject, if the subject is not same program is looking for the book name and search the book if there is, another wise book is not found. For example we want to find the book by author, steps are:

1. Click the Author button
2. Enter the Author name
3. Press the Enter key to find the book.

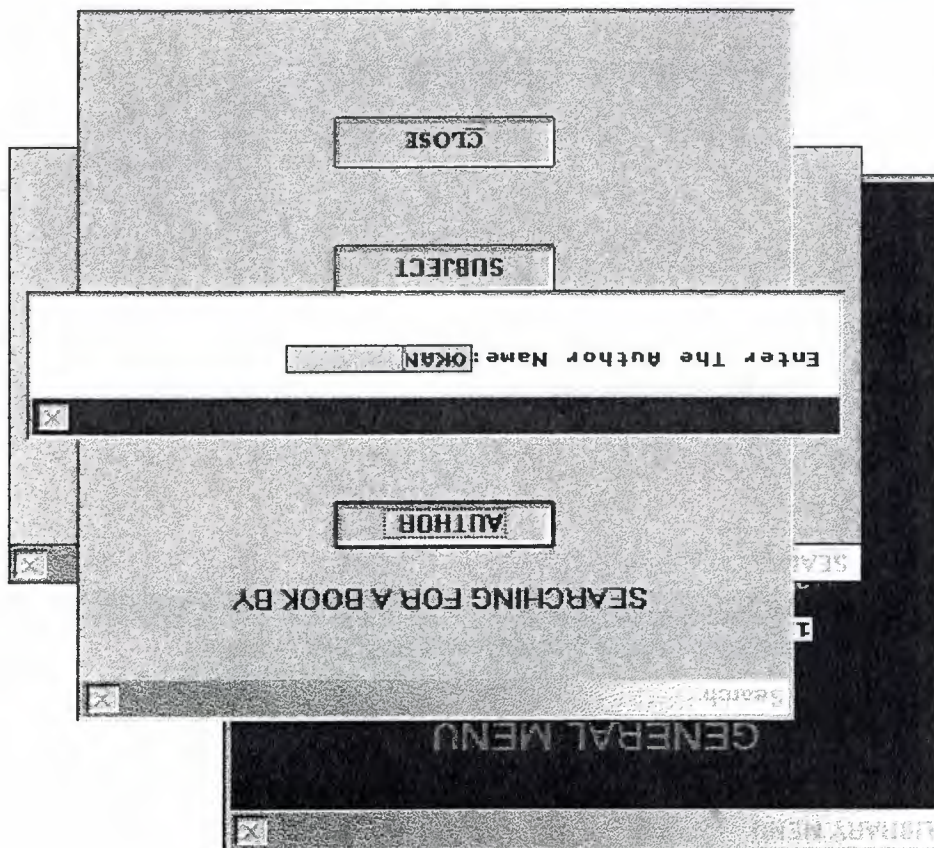
The screens of the searching a book by Author are shown as 4. *SEARCH A BOOK* in 3 steps.



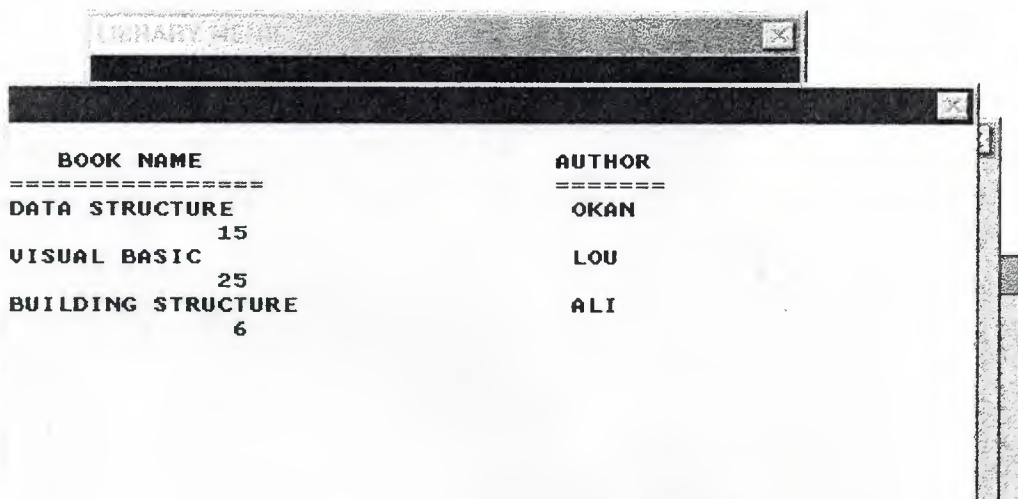




Search			
Book_name	Serial_no	Author	Subject
USUAL BASIC	25 LOU		PROGRAM
DATA STRUCTURE	15 OKAN		STRUCTUI
FOCUS ON GRAMMER	2 DAVID		LANGUAGI
BUSINESS INFORMATION	5 4		INTRODUC
BUILDING STRUCTURE	6 ALI		INFORMAT
SIGNAL AND SYSTEM	8 FAHRADDIN		SYSTEM
DATA STRUCTURE	15 OKAN		STRUCTUI

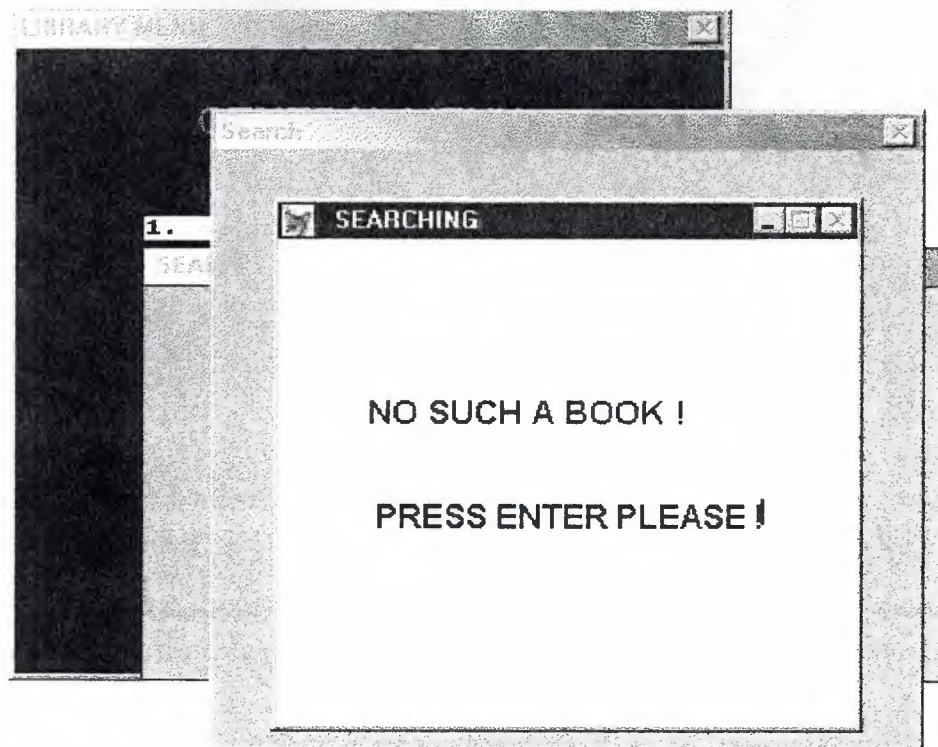


If the subject we are looking for is not same as the subject we have in our database, then the program can search for books name. If it's found all the book will be listed in another screen which is look like below. Here we are search the books by later C, program can look first to the subject, if not such a subject is available, then it will look to the book name and bring all the books include the later C.



BOOK NAME	AUTHOR
DATA STRUCTURE	OKAN
15	
VISUAL BASIC	LOU
25	
BUILDING STRUCTURE	ALI
6	

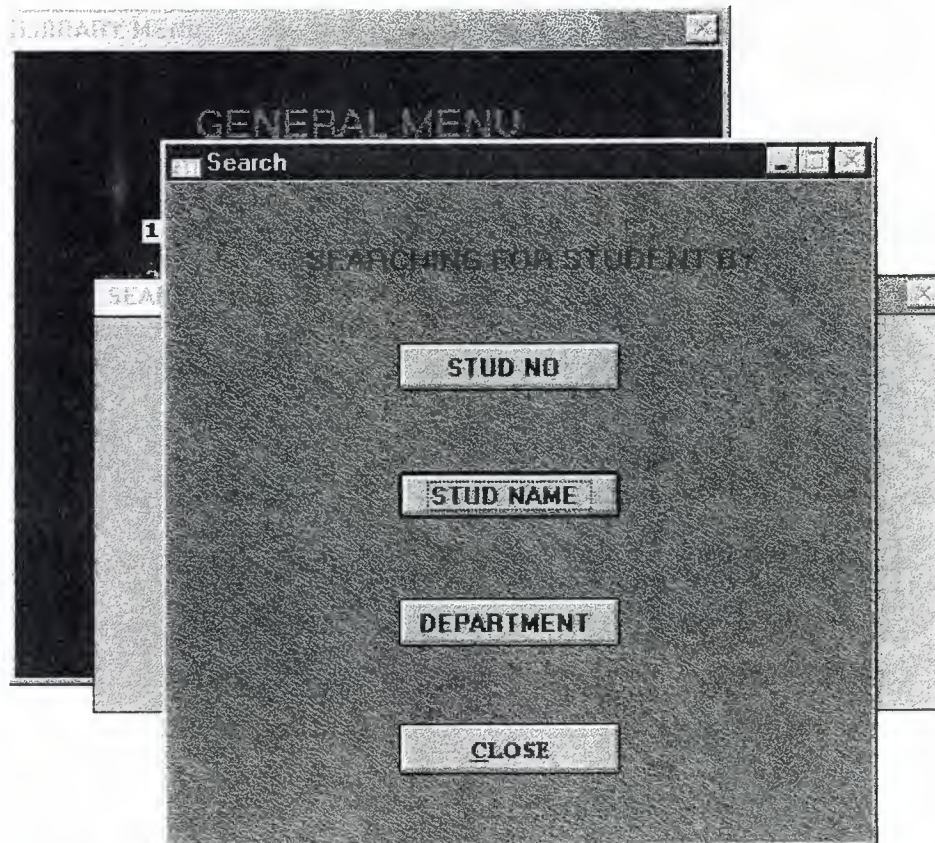
If the book not found, it will give the following message.

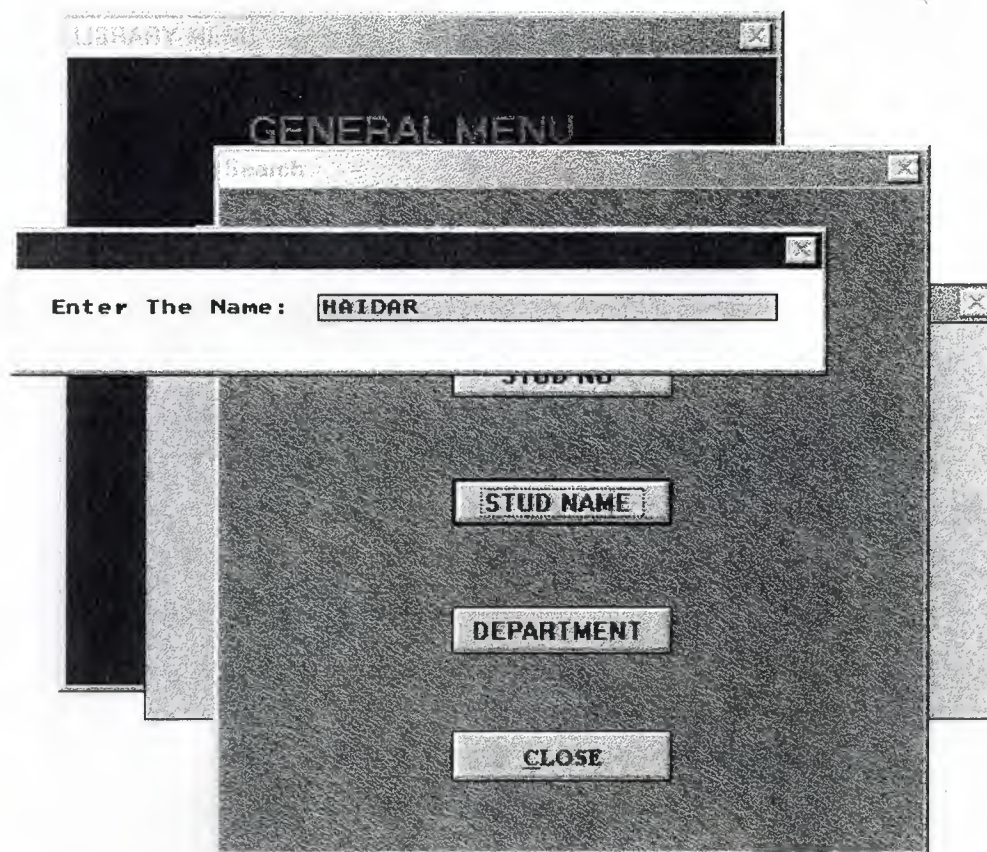




## 5. Searching For A Student

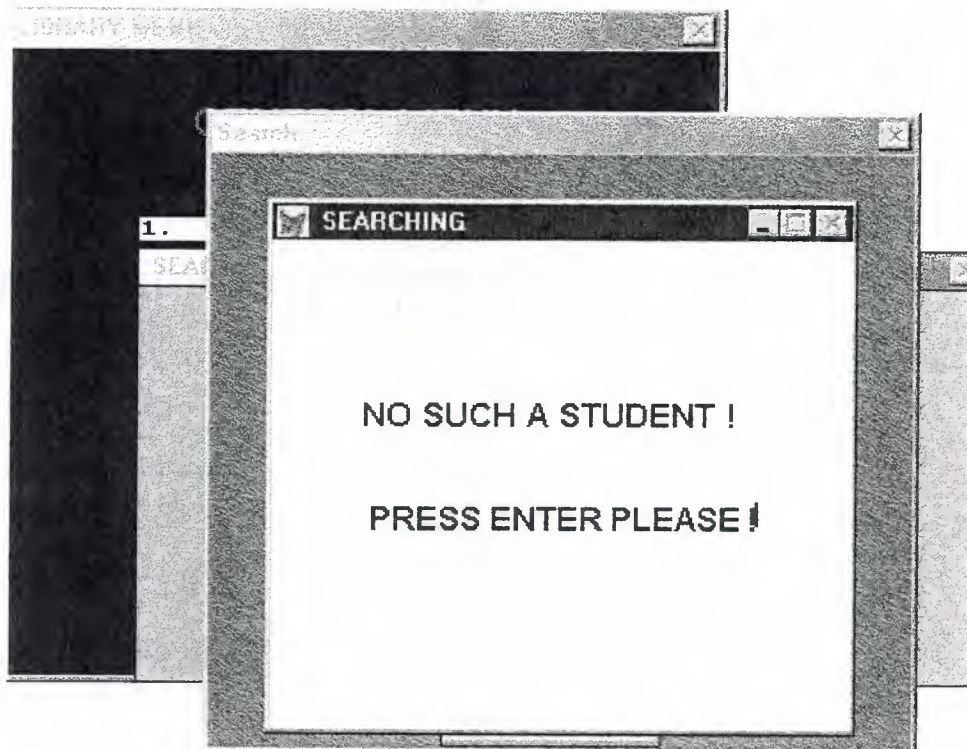
Again, in the general searching menu we are going to search for the student. In the searching menu, if we press the *STUDENT* button we can see the screen, which includes the Student Number, Student Name and Department, and the screens are shown as 5. *Searching For A Student* in three steps







In the searching for student, if the student could not be found by Student\_ID, Student Name and student department. Then the program shows the following message (No such a student! Press enter please!!). which is shown as follow:





```

*      *      BOOK12/Windows Databases, Indexes, Relations
*      *
*      *****

IF USED("c:\mak\kitab.dbf")
    SELECT kitab
    *SET ORDER TO TAG "bn"
ELSE
    SELECT 0
    USE (LOCFILE("c:\mak\kitab.dbf","DBF","Where is kitab?"));
    AGAIN ALIAS kitab ;
    ORDER TAG "bn"
ENDIF

IF USED("c:\mak\talib.dbf")
    SELECT talib;
    SET ORDER TO TAG "nam"
ELSE
    SELECT 0
    USE (LOCFILE("c:\mak\talib.dbf","DBF","Where is talib?"));
    AGAIN ALIAS talib;
    ORDER TAG "nam"
*ENDIF

*      *****
*      *
*      *      Windows Window definitions
*      *
*      *****

IF NOT WEXIST("_rqo1bm9wq")
    DEFINE WINDOW _rqo1bm9wq ;
        AT 0.000, 0.000 ;
        SIZE 15.075, 70.333 ;
        TITLE " SEARCHING " ;
        FONT "MS Sans Serif", 8 ;
        STYLE "B" ;
        FLOAT ;
        CLOSE ;
    MINIMIZE ;
    COLOR RGB(,,192,192,192)
    MOVE WINDOW _rqo1bm9wq CENTER
ENDIF

*      *****

```

```

*      *
*      *      BOOK12/Windows Setup Code - SECTION 2
*      *
*      *****

```

```

MOVE WINDOW '_rqo1bm9wq' CENTER
PRIVATE isediting, isadding, wztblarr
PRIVATE wzolddelete, wzolderror, wzoldesc
PRIVATE wzalias, tempcurs, wzlastrec
PRIVATE isreadonly, find_drop, is2table
IF EMPTY(ALIAS())
    WAIT WINDOW C_NOTABLE
    RETURN
ENDIF

```

```

m.wztblarr= "
m.wzalias=SELECT()
m.isediting=.F.
m.isadding=.F.
m.is2table = .F.
m.wzolddelete=SET('DELETE')
SET DELETED ON
m.tempcurs=SYS(2015) &&used if General field
m.wzlastrec = 1
m.wzolderror=ON('error')
ON ERROR DO wizerrorhandler
wzoldesc=ON('KEY','ESCAPE')
ON KEY LABEL ESCAPE
m.find_drop = IIF(_DOS,0,2)
m.isreadonly=IIF(ISREAD(),.T.,.F.)
IF m.isreadonly
    WAIT WINDOW C_READONLY TIMEOUT 1
ENDIF

```

```

IF RECCOUNT()=0 AND !m.isreadonly AND fox_alert(C_DBFEMPTY)
    APPEND BLANK
ENDIF
GOTO TOP
SCATTER MEMVAR MEMO

```

```

*      *****
*      *
*      *      BOOK12/Windows Screen Layout

```



```
*
*
* *****
```

```
#REGION 1
```

```
IF WVISIBLE("_rqolbm9wq")
```

```
    ACTIVATE WINDOW _rqolbm9wq SAME
```

```
ELSE
```

```
    ACTIVATE WINDOW _rqolbm9wq NOSHOW
```

```
ENDIF
```

```
@ 1.923,12.000 SAY "  CHOOCE A SEARCH TOPIC:" ;
```

```
    FONT "MS Sans Serif", 10 ;
```

```
    STYLE "BT"
```

```
@ 4.845,10.667 GET d2 ;
```

```
    PICTURE "@*HN \<BOOK" ;
```

```
    SIZE 2.0,12.00,0.667 ;
```

```
    DEFAULT 1 ;
```

```
    FONT "times new roman", 14 ;
```

```
    STYLE "B" ;
```

```
    VALID bt ('STACK');
```

```
    MESSAGE 'Looking for a book'
```

```
@ 4.845,40.667 GET d2 ;
```

```
    PICTURE "@*HN \<STUDENT" ;
```

```
    SIZE 2.0,10.0,0.667 ;
```

```
    DEFAULT 1 ;
```

```
    FONT "times new roman", 14 ;
```

```
    STYLE "B" ;
```

```
    VALID bt ('LOCATE');
```

```
    MESSAGE 'Searching for student data.'
```

```
@ 9.845,25.667 GET d2 ;
```

```
    PICTURE "@*HN \<CLOSE" ;
```

```
    SIZE 2.0,12.0,0.667 ;
```

```
    DEFAULT 1 ;
```

```
    FONT "times new roman", 14 ;
```

```
    STYLE "B" ;
```

```
    VALID bt('EXIT1');
```

```
    MESSAGE 'Close screen.'
```

```
IF NOT WVISIBLE("_rqolbm9wq")
```

```
    ACTIVATE WINDOW _rqolbm9wq
```

```
ENDIF
```

```

*      *****
*
*      *   WindowsREAD contains clauses from SCREEN book12
*      *
*      *****

```

```

READ CYCLE ;
    ACTIVATE READACT() ;
    DEACTIVATE READDEAC() ;
    NOLOCK
RELEASE WINDOW _rqo1bm9wq

```

```

*      *****
*
*      *   Windows Closing Databases
*      *
*      *****

```

```

IF USED("c:\mak\kitab.dbf")
    SELECT kitab
    USE
ENDIF

```

```

IF USED("c:\mak\talib.dbf")
    SELECT talib
    USE
ENDIF
SELECT (m.currarea)

```

```

#REGION 0
SET READBORDER &rborder
IF m.talkstat = "ON"
    SET TALK ON
ENDIF
IF m.compstat = "ON"
    SET COMPATIBLE ON
ENDIF

```

```

*      *****
*
*      *   BOOK12/Windows Cleanup Code
*      *
*      *****

```

```

#REGION 1
SET DELETED &wzolddelete
SET FIELDS &wzfields
ON ERROR &wzolderror
ON KEY LABEL ESCAPE &wzoldesc
DO CASE
CASE _DOS AND SET('DISPLAY')='VGA25'
    @24,0 CLEAR TO 24,79
CASE _DOS AND SET('DISPLAY')='VGA50'
    @49,0 CLEAR TO 49,79
CASE _DOS
    @24,0 CLEAR TO 24,79
ENDCASE

```

\*\*\*\*Procedures\*\*\*\*

```

* *****
*
* * BOOK12/Windows Supporting Procedures and Functions
*
* *****

```

```

#REGION 1
PROCEDURE readdeac
IF isediting
    ACTIVATE WINDOW '_rqo1bm9wq'
    WAIT WINDOW C_EDIT$ NOWAIT
ENDIF
IF !WVISIBLE(WOUTPUT())
    CLEAR READ
    RETURN .T.
ENDIF
RETURN .F.

```

```

PROCEDURE readact
IF lisediting
    SELECT (m.wzalias)
    SHOW GETS
ENDIF
DO REFRESH
RETURN

```

```

PROCEDURE wizerrorhandler
    WAIT WINDOW message()

```

RETURN

#### PROCEDURE BTN\_VAL

```
PARAMETER m.btnname
DO CASE
CASE m.btnname='LOCATE'
    DO MEXXSS
CASE m.btnname='STAK'
    DO DONADONI
CASE m.btnname='EXIT'
    m.bailout=.T. &&this is needed if used with FoxApp
    CLEAR READ
    RETURN
ENDCASE
SCATTER MEMVAR MEMO
SHOW GETS
```

RETURN

#### PROCEDURE REFRESH

```
DO CASE
CASE m.isreadonly AND RECCOUNT()=0
    SHOW GETS DISABLE
    SHOW GET exit_btn ENABLE
CASE m.isreadonly
    SHOW GET add_btn DISABLE
    SHOW GET del_btn DISABLE
    SHOW GET edit_btn DISABLE
CASE (RECCOUNT()=0 OR EOF()) AND !m.isediting
    SHOW GETS DISABLE
    SHOW GET add_btn ENABLE
    SHOW GET exit_btn ENABLE
CASE m.isediting
    SHOW GET find_drop DISABLE
    SHOW GET top_btn DISABLE
    SHOW GET prev_btn DISABLE
    SHOW GET loc_btn ENABLE
    SHOW GET next_btn DISABLE
    SHOW GET end_btn DISABLE
    SHOW GET add_btn DISABLE
    SHOW GET prnt_btn DISABLE
    SHOW GET exit_btn DISABLE
    SHOW GET edit_btn,1 PROMPT "<Save"
    SHOW GET del_btn,1 PROMPT "<Cancel"
    ON KEY LABEL ESCAPE DO BTN_VAL WITH 'DELETE'
RETURN
```



```

OTHERWISE
    SHOW GET edit_btn,1 PROMPT "Ed\<it"
    SHOW GET del_btn,1 PROMPT "\<Delete"
    SHOW GETS ENABLE
ENDCASE
IF m.is2table
    SHOW GET add_btn DISABLE
ENDIF
ON KEY LABEL ESCAPE
RETURN

```

\*\*\*\*\*

#### PROCEDURE MEXXSS

```

    DEFINE WINDOW _qjn12T ;
        AT 0.000,0.000 ;
        SIZE 20.615,50 ;
        TITLE "Search" ;
        FONT "MS Sans Serif", 10 ;
        COLOR RGB(,,192,192,192);
        FLOAT CLOSE MINIMIZE SYSTEM
    MOVE WINDOW _qjn12T CENTER
    ACTIVATE WINDOW _qjn12T NOSHOWN

@ 4.00,16.500 GET d1 ;
    PICTURE "@*HN AUTHOR " ;
    SIZE 1.846,18.333,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    VALID nimra ()

@ 8.00,16.500 GET d1 ;
    PICTURE "@*HN BOOK NAME " ;
    SIZE 1.846,18.333,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    VALID isim ()

@ 12.00,16.500 GET d1 ;
    PICTURE "@*HN SUBJECT " ;
    SIZE 1.846,18.333,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;

```

```

STYLE "B" ;
VALID mada ()

@ 15.845,16.5 GET m.exit_btn ;
  PICTURE "@*HN \<CLOSE" ;
  SIZE 2.0,22.0,0.667 ;
  DEFAULT 1 ;
  FONT "times new roman", 8 ;
  STYLE "B" ;
  VALID btn_val('EXIT') ;
  MESSAGE 'Close screen.'

@ 1.923,10.000 SAY "SEARCHING FOR A BOOK BY" ;
  FONT "MS Sans Serif", 10 ;
  STYLE "BT"
  *COLOR RGB(,,192,192,192)

ACTIVATE WINDOW _qjn12T
  READ
  RELEASE WINDOW _qjn12T
RETURN

*****
PROCEDURE MEXXSS1
  DEFINE WINDOW _qjn112 ;
    AT 0.000, 0.000 ;
    SIZE 18.615,57.667 ;
    TITLE " SEARCHING " ;
    FONT "MS Sans Serif", 8 ;
    FLOAT NOCLOSE MINIMIZE SYSTEM
  MOVE WINDOW _qjn112 CENTER
  ACTIVATE WINDOW _qjn112 NOSHOWN
@ 5.923,7.000 SAY " NO SUCH A BOOK ! " ;
FONT "Arial", 13 ;
  STYLE "BT" ;
  PICTURE "@J" ;
  COLOR RGB(,,0,128,128)
@ 9.923,9.000 SAY "PRESS ENTER PLEASE ! " ;
FONT "Arial", 12 ;
  STYLE "BT" ;
  PICTURE "@J" ;
  COLOR RGB(,,0,128,128)
ACTIVATE WINDOW _qjn112
  READ
  RELEASE WINDOW _qjn112

```

RETURN

PROCEDURE MEXXSS2

```
    DEFINE WINDOW _qjn112 ;
        AT 0.000, 0.000 ;
        SIZE 18.615, 57.667 ;
        TITLE " SEARCHING " ;
        FONT "MS Sans Serif", 8 ;
        FLOAT NOCLOSE MINIMIZE SYSTEM
    MOVE WINDOW _qjn112 CENTER
    ACTIVATE WINDOW _qjn112 NOSHOWN
@ 5.923, 7.000 SAY " NO SUCH A STUDENT ! " ;
FONT "Arial", 13 ;
    STYLE "BT" ;
    PICTURE "@J" ;
    COLOR RGB(,,, 0, 128, 128)
@ 9.923, 9.000 SAY "PRESS ENTER PLEASE ! " ;
FONT "Arial", 12 ;
    STYLE "BT" ;
    PICTURE "@J" ;
    COLOR RGB(,,, 0, 128, 128)
ACTIVATE WINDOW _qjn112
    READ
    RELEASE WINDOW _qjn112
RETURN
```

CLOSE DATABASE

\*\*\*\*\* SEARCHING BY BOOK NAME \*\*\*\*\*

procedure isim

```
    CLOSE DATABASE
    DEFINE WINDOW aW1 AT 12, 12 SIZE 4, 50 SYSTEM
    ACTIVATE WINDOW aW1
    your_ch = " "
    @ 1, 1 say " Enter The Name: "
    @ 1, 19 get your_ch
    READ
    RELEASE WINDOW aW1
    use \mak\kitab.dbf
    SET ORDER TO TAG "Bn"
*IF sel=1
    LOCATE ALL FOR UPPER(ALLTRIM(book_name)) = UPPER(ALLTRIM(your_ch))
    IF FOUND()
    *THEN
    *CLOSE ALL
    WAIT WINDOW " Searching is OK " NOWAIT
    DEFINE WINDOW aa1 ;
```

```

        AT 0.000,0.000 ;
        SIZE 37.075,120.333 ;
        TITLE "Search" ;
        FONT "MS Sans Serif", 8 ;
        STYLE "B" ;
        FLOAT ;
        CLOSE ;
    MINIMIZE
        *COLOR RGB(,,192,192,192)
        MOVE WINDOW aa1 CENTER
        activate window aa1
    BROWSE WINDOW AA1
    deactivate window aa1
    * ENDIF
    *CONTINUE
    ELSE
        WAIT WINDOW " IT'S NOT FOUND " NOWAIT
    DO MEXXSS1
    *ENDIF
ENDIF
return
***** SEARCHING BY BOOK AUTHOR *****
procedure nimra
CLOSE DATABASE
DEFINE WINDOW a1 AT 12,12 SIZE 4,50 SYSTEM
    MOVE WINDOW A1 CENTER
    ACTIVATE WINDOW a1
    your_ch=" "
    @1,1 say "Enter The Author Name: "
    @1,23 get your_ch
    READ
    RELEASE WINDOW a1
    use c:\mak\kitab.dbf
    *SET ORDER TO TAG "nn"
    LOCATE ALL FOR UPPER(ALLTRIM(author))= UPPER(ALLTRIM(your_ch))
    IF FOUND() THEN
        WAIT WINDOW " Searching is OK " NOWAIT
        DEFINE WINDOW aa1 ;
            AT 0.000,0.000 ;
            SIZE 37.075,120.333 ;
            TITLE "Search" ;
            FONT "MS Sans Serif", 8 ;
            STYLE "B" ;
            FLOAT ;
            CLOSE ;
        MINIMIZE

```



```

        MOVE WINDOW aa1 CENTER
        activate window aa1
    BROWSE WINDOW AA1
    deactivate window aa1
    ELSE
        WAIT WINDOW " IT'S NOT FOUND " NOWAIT
    DO MEXXSS1
ENDIF
return

```

\*\*\*\*\* SEARCHING BY BOOK SUBJECT \*\*\*\*\*

procedure mada

```

    CLOSE DATABASE
    DEFINE WINDOW aQ1 AT 12,12 SIZE 4,50 SYSTEM
    ACTIVATE WINDOW aQ1
    your_ch=" "
    @1,1 say " Enter The Subject: "
    @1,21 get your_ch
    READ
    RELEASE WINDOW aQ1
    use \mak\kitab.dbf
    SET ORDER TO TAG "Su"

```

```

*DEFINE WINDOW a2 AT 12,12 SIZE 40,50 SYSTEM

```

```

*ACTIVATE WINDOW a2

```

```

*? ' Name      author.....'

```

```

*? '=====

```

```

LOCATE ALL FOR UPPER(ALLTRIM(subject))=UPPER(ALLTRIM(your_ch))
IF NOT Found()

```

```

    DEFINE WINDOW a2 AT 6,10 SIZE 30,60 SYSTEM

```

```

    ACTIVATE WINDOW a2

```

```

* ? subject

```

```

*else

```

```

    ?' BOOK NAME          AUTHOR '

```

```

    ?'=====            =====

```

```

    For c=Len(Book_name) to len(Your_ch) STEP -1

```

```

        LOCATE ALL FOR

```

```

UPPER(ALLTRIM(Right(book_name,(c))))=UPPER(ALLTRIM(your_ch))

```

```

        IF Found()

```

```

            ? Book_Name

```

```

            ?? author

```

```

            ?? kitab.serial_no

```

```

        ENDIF

```

```

    EndFor

```

```

Wait Window 'Press Any Key to Continue...'
Deactivate window a2
*DEACTIVE PROCEDURE MEXXSS1
ENDIF
IF FOUND()
WAIT WINDOW " Searching is OK " NOWAIT
DEFINE WINDOW aal ;
    AT 0.000,0.000 ;
    SIZE 37.075,120.333 ;
    TITLE " " ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    FLOAT ;
    CLOSE ;
MINIMIZE
    MOVE WINDOW aal CENTER
    activate window aal
BROWSE WINDOW AA1
deactivate window aal
    ELSE
    deactivate window aal
    *DO MEXXSS1
    *WAIT WINDOW " No such a book " NOWAIT
ENDIF

return

```

\*\*\*\*\*

```

PROCEDURE DONADONI
    DEFINE WINDOW _qjn121 ;
        AT 0.000,0.000 ;
        SIZE 20.615,50 ;
        TITLE "Search" ;
        FONT "MS Sans Serif", 10 ;
        COLOR RGB(,,100,100,122);
        FLOAT CLOSE MINIMIZE SYSTEM
    MOVE WINDOW _qjn121 CENTER
    ACTIVATE WINDOW _qjn121 NOSHOW

```

```

@ 5.00,16.500 GET d1 ;
    PICTURE "@*HN STUD NO " ;
    SIZE 1.846,18.333,0.667 ;

```

```
DEFAULT 1 ;  
FONT "MS Sans Serif", 8 ;  
STYLE "B" ;  
VALID star1 ()
```

```
@ 9.00,16.500 GET d1 ;  
  PICTURE "@*HN STUD NAME " ;  
  SIZE 1.846,18.333,0.667 ;  
  DEFAULT 1 ;  
  FONT "MS Sans Serif", 8 ;  
  STYLE "B" ;  
  VALID isim1 ()
```

```
@ 13.00,16.500 GET d1 ;  
  PICTURE "@*HN DEPARTMENT " ;  
  SIZE 1.846,18.333,0.667 ;  
  DEFAULT 1 ;  
  FONT "MS Sans Serif", 8 ;  
  STYLE "B" ;  
  VALID USRA ()
```

```
@ 16.845,16.5 GET m.exit_btn ;  
  PICTURE "@*HN \<CLOSE" ;  
  SIZE 2.0,22.0,0.667 ;  
  DEFAULT 1 ;  
  FONT "times new roman", 8 ;  
  STYLE "B" ;  
  VALID btn_val('EXIT') ;  
  MESSAGE 'Close screen.'
```

```
@ 1.923,10.000 SAY "SEARCHING FOR STUDENT BY" ;  
  FONT "MS Sans Serif", 10 ;  
  STYLE "BT";  
  COLOR RGB(0,233,1,192,192,192)  
ACTIVATE WINDOW _qjn121  
  READ  
  RELEASE WINDOW _qjn121  
RETURN
```

```
*IF USED("c:\mak\talib.dbf")  
*   SELECT talib;  
*   SET ORDER TO TAG "nam"  
*ELSE  
*   SELECT 0  
*   USE (LOCFILE("c:\mak\talib.dbf","DBF","Where is talib?"));
```



```

*          AGAIN ALIAS talib
*          ORDER TAG "nam"
*ENDIF
*****
***** SEARCHING BY Student NAME *****
procedure isim1
    CLOSE DATABASE
    DEFINE WINDOW B1 AT 12,12 SIZE 4,50 SYSTEM
    ACTIVATE WINDOW B1
    your_ch="          "
    @1,1 say " Enter The Name:  "
    @1,19 get your_ch
    READ
    RELEASE WINDOW B1

    use \mak\talib.dbf
    SET ORDER TO TAG "NAM"
*IF sel=1
    LOCATE ALL FOR UPPER(ALLTRIM(first_name))= UPPER(ALLTRIM(your_ch))
    IF FOUND()
    *USE kitab
    *THEN
    *CLOSE ALL
    WAIT WINDOW " Searching is OK " NOWAIT
    DEFINE WINDOW aa11 ;
        AT 0.000,0.000 ;
        SIZE 37.075,120.333 ;
        TITLE "N.E.U" ;
        FONT "MS Sans Serif", 8 ;
        STYLE "B" ;
        FLOAT ;
        CLOSE ;
    MINIMIZE
    *COLOR RGB(,,192,192,192)
    MOVE WINDOW aa11 CENTER
    activate window aa11
    BROWSE WINDOW AA11
    deactivate window aa11
    * ENDIF
    ELSE
    WAIT WINDOW " IT'S NOT FOUND " NOWAIT
    DO MEXXSS2
    *ENDIF
ENDIF
return
***** SEARCHING BY STUDENT NUMBER *****

```

procedure star1

```
    CLOSE DATABASE
    DEFINE WINDOW axx1 AT 12,12 SIZE 4,50 SYSTEM
    ACTIVATE WINDOW axx1
    your_ch="      "
    @1,1 say " Enter Student No:  "
    @1,21 get your_ch
    READ
    RELEASE WINDOW axx1
    use \mak\talib.dbf
    SET ORDER TO TAG "std"
    LOCATE ALL FOR UPPER(ALLTRIM(st_id))= UPPER(ALLTRIM(your_ch))
    IF FOUND()
        WAIT WINDOW " Searching is OK " NOWAIT
        DEFINE WINDOW aaxx1 ;
            AT 0.000, 0.000 ;
            SIZE 37.075,120.333 ;
            TITLE " SEARCHING " ;
            FONT "MS Sans Serif", 8 ;
            STYLE "B" ;
            FLOAT ;
            CLOSE ;
        MINIMIZE
        MOVE WINDOW aaxx1 CENTER
        activate window aaxx1
    BROWSE WINDOW aaxx1
    deactivate window aaxx1
    ELSE
        DO MEXXSS2
    ENDIF
```

return

\*\*\*\*\* SEARCHING BY DEPARTMENT \*\*\*\*\*

procedure USRA

```
    CLOSE DATABASE
    DEFINE WINDOW ax1 AT 12,12 SIZE 4,50 SYSTEM
    ACTIVATE WINDOW ax1
    your_ch="      "
    @1,1 say " Enter Dep. Name:  "
    @1,21 get your_ch
    READ
    RELEASE WINDOW ax1
    use \mak\talib.dbf
    SET ORDER TO TAG "dep"
    LOCATE ALL FOR UPPER(ALLTRIM(department))= UPPER(ALLTRIM(your_ch))
```

```

IF FOUND()
  WAIT WINDOW " Searching is OK " NOWAIT
  DEFINE WINDOW aax1 ;
    AT 0.000,0.000 ;
    SIZE 37.075,120.333 ;
    TITLE " SEARCHING " ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    FLOAT ;
    CLOSE ;
  MINIMIZE
    MOVE WINDOW aax1 CENTER
    activate window aax1
  BROWSE WINDOW AAx1
  deactivate window aax1
  ELSE
    DO MEXXSS2
  ENDIF

```

return

```

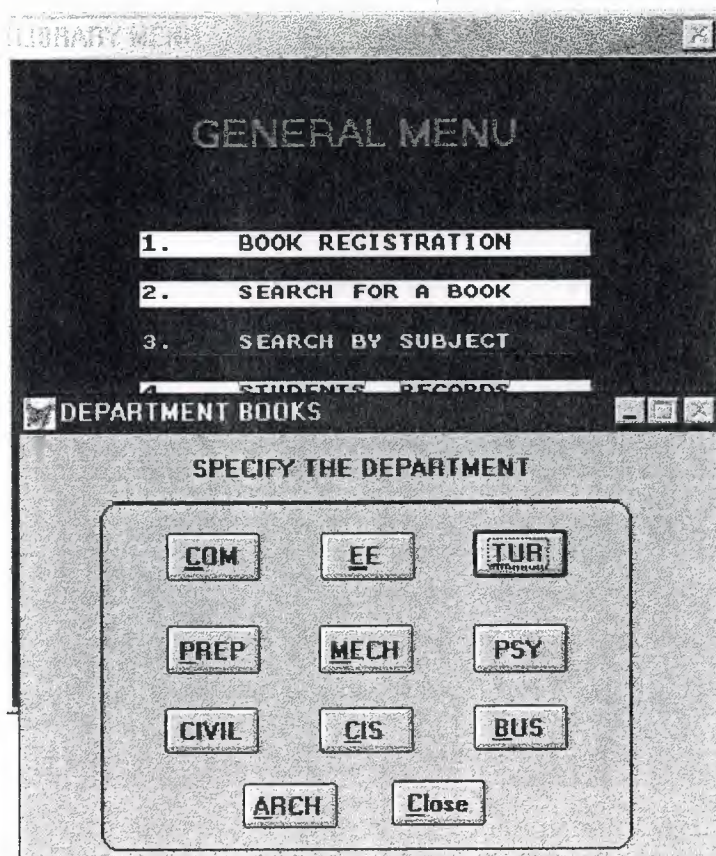
PROCEDURE BT
  PARAMETER m.btnname
  DO CASE
CASE m.btnname='STACK'
  DO MEXXSS
CASE m.btnname='LOCATE'
  DO DONADONI
CASE m.btnname='EXIT1'
  set sysm to defa
  DEACTIVATE WINDOW _rqolbm9wq
  RELEASE WINDOW _rqolbm9wq
  SET CLOCK OFF
  ENDCASE
RETURN

```



## 6. Searching By Subject (Department)

In the screen of searching for a subject we are specify the books according to their departments. It is more useful for those that they don't have any idea about the books of one departments, he/she can press the specific icons to get all the books of that departments. The screen is shown as 6. *Searching By Subject*.



LIBRARY SYSTEM

## GENERAL MENU

1. BOOK REGISTRATION
2. SEARCH FOR A BOOK
3. SEARCH BY SUBJECT
4. STUDENTS RECORDS

DEPARTMENT

### SPECIFY THE DEPARTMENT

DEPARTMENT BOOKS

Book_name	Serial_no
BUSINESS ADMINISTRATION	
BUSINESS INFORMATION	

## Query of searching by department

Query of Architecture

```
SELECT *;  
FROM Kitab;  
WHERE Kitab.book_type = "ARCH"
```

Query of Business

```
SELECT *;  
FROM Kitab;  
WHERE Kitab.book_type = "BUS"
```

Query of Computer Information System

```
SELECT *;  
FROM Kitab;  
WHERE Kitab.book_type = "CIS"SELECT *;
```

Query of Civil

```
SELECT *;  
FROM Kitab;  
WHERE Kitab.book_type = "CIVIL"
```

Query of Computer

```
SELECT *;  
FROM Kitab;  
WHERE Kitab.book_type = "COM"
```

Query of Electrical Engineering

```
SELECT *;  
FROM Kitab;  
WHERE Kitab.book_type = "EE"
```

Query of Mechanical

```
SELECT *;  
FROM Kitab;  
WHERE Kitab.book_type = "MECH"
```

Query of Physical

```
SELECT *;  
FROM Kitab;  
WHERE Kitab.book_type = "PSY"
```



Query of Prep

```
SELECT *;  
FROM Kitab;  
WHERE Kitab.book_type = "PREP"
```

Query of Truism

```
SELECT *;  
FROM Kitab;  
WHERE Kitab.book_type = "TUR"
```

## 7. Student Information

This screen gives the necessary information about the student that is enough in order to be a member in the library. It can Add and Delete the records that contain the fields like Student ID, Name, Last Name, Department, nationality and class. The screen is shown as 7. *Student Record*

LIBRARY MENU

GENERAL MENU

BOOK REGISTRATION

**STUDENT INFORMATION**

Student Id: 960588

First Name: HAIDAR

Last Name: DOZANDEH

Department: COM

Nationality: IRAN

Class: 4TH

Top Prev Next End Locate Add Edit Delete Print Close



## Program how to create the student screen

```
* *****
*
* * 23.12.99          TALIB.SPR          15:15:26
*
* *****
*
* * Author's Name
*
* * Copyright (c) 1999 Company Name
* * Address
* * City,   Zip
*
* * Description:
* * This program was automatically generated by GENSCRN.
*
* *****
```

```
* *****
*
* *      TALIB/Windows Setup Code - SECTION 1
*
* *****
*
```

```
#REGION 1
PRIVATE wzfields,wztalk
IF SET("TALK") = "ON"
    SET TALK OFF
    m.wztalk = "ON"
ELSE
    m.wztalk = "OFF"
ENDIF
m.wzfields=SET('FIELDS')
SET FIELDS OFF
IF m.wztalk = "ON"
    SET TALK ON
ENDIF
```

```
#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
```



```

        SET TALK OFF
        m.talkstat = "ON"
ELSE
        m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.rborder = SET("READBORDER")
SET READBORDER ON

m.currarea = SELECT()

* *****
*
*   TALIB/Windows Databases, Indexes, Relations
*
* *****
*

IF USED("talib")
    SELECT talib
    SET ORDER TO TAG "std"
ELSE
    SELECT 0
    USE (LOCFILE("talib.dbf", "DBF", "Where is talib?"));
    AGAIN ALIAS talib ;
    ORDER TAG "std"
ENDIF

* *****
*
*   Windows Window definitions
*
* *****
*

IF NOT WEXIST("_sb60wp9tq")
    DEFINE WINDOW _sb60wp9tq ;
        AT 0.000, 0.000 ;
        SIZE 18.692, 83.333 ;
        FONT "MS Sans Serif", 8 ;
        STYLE "B" ;
        FLOAT ;

```

```

        CLOSE ;
        MINIMIZE ;
        COLOR RGB(,,192,192,192)
        MOVE WINDOW _sb60wp9tq CENTER
    ENDIF

```

```

* *****
*
*      TALIB/Windows Setup Code - SECTION 2
*
* *****
*

```

```

#REGION 1

```

```

#DEFINE C_DBFEMPTY      'Database is empty, add a record?'
#DEFINE C_EDITS          'Please finish your edits.'
#DEFINE C_TOPFILE        'Top of file.'
#DEFINE C_ENDFILE        'End of file.'
#DEFINE C_BRTITLE        'Locate Record'
#DEFINE C_NOLOCK         'Sorry, could not lock record -- try again later.'
#DEFINE C_ECANCEL        'Edits Canceled.'
#DEFINE C_DELREC         'Delete selected record?'
#DEFINE C_NOFEAT         'Feature not available yet.'
#DEFINE C_NOWIZ          'Wizard application is not available.'
#DEFINE C_MAKEREPO       'Creating report with Report Wizard.'
#DEFINE C_NOREPO         'Could not create report.'
#DEFINE C_DELNOTE        'Deleting records...'
#DEFINE C_READONLY       'Table is read-only. No editing allowed.'
#DEFINE C_NOTABLE        'No table selected. Open table or run query.'
#DEFINE C_BADEXP         'Invalid expression.'
#DEFINE C_LOCWIZ         'Locate WIZARD.APP:'
#DEFINE C_MULTITABLE     'You have multiple related tables. Adding records in not
allowed.'

```

```

MOVE WINDOW '_sb60wp9tq' CENTER
PRIVATE isediting,isadding,wztblarr
PRIVATE wzolddelete,wzolderror,wzoldesc
PRIVATE wzalias, tempcurs,wzlastrec
PRIVATE isreadonly,find_drop,is2table

```

```

IF EMPTY(ALIAS())
    WAIT WINDOW C_NOTABLE
    RETURN
ENDIF

```

```

m.wztblarr= "
m.wzalias=SELECT()
m.isediting=.F.
m.isadding=.F.
m.is2table = .F.
m.wzolddelete=SET('DELETE')
SET DELETED ON
m.tempcurs=SYS(2015) &&used if General field
m.wzlastrec = 1
m.wzolderror=ON('error')
ON ERROR DO wizerrorhandler
wzoldesc=ON('KEY','ESCAPE')
ON KEY LABEL ESCAPE
m.find_drop = IIF(_DOS,0,2)

m.isreadonly=IIF(ISREAD(),.T.,.F.)
IF m.isreadonly
    WAIT WINDOW C_READONLY TIMEOUT 1
ENDIF

```

```

IF RECCOUNT()=0 AND !m.isreadonly AND fox_alert(C_DBFEMPTY)
    APPEND BLANK
ENDIF

```

```

GOTO TOP
SCATTER MEMVAR MEMO

```

```

*      *****
*      *
*      *      TALIB/Windows Screen Layout
*      *
*      *****
*

```

```

#REGION 1
IF WVISIBLE("_sb60wp9tq")
    ACTIVATE WINDOW _sb60wp9tq SAME
ELSE
    ACTIVATE WINDOW _sb60wp9tq NOSHOW
ENDIF
@ 2.692,0.000 TO 2.692,83.333 ;
    PEN 2, 8 ;
    STYLE "1"
@ 3.692,18.000 SAY "Student Id:" ;

```

```

SIZE 0.938,13.500 ;
FONT "Times New Roman", 10 ;
STYLE "BT" ;
PICTURE "@J" ;
COLOR RGB(,,,255,255,255)
@ 3.769,34.333 GET m.st_id ;
SIZE 1.000,12.000 ;
DEFAULT " " ;
FONT "Times New Roman", 8 ;
STYLE "B" ;
PICTURE "@K XXXXXXXXXXXX" ;
WHEN isediting ;
COLOR ,RGB(0,0,0,255,255,255)
@ 5.538,18.000 SAY "First Name:" ;
SIZE 0.938,13.500 ;
FONT "Times New Roman", 10 ;
STYLE "BT" ;
PICTURE "@J" ;
COLOR RGB(,,,255,255,255)
@ 5.615,34.333 GET m.first_name ;
SIZE 1.000,28.000 ;
DEFAULT " " ;
FONT "Times New Roman", 8 ;
STYLE "B" ;
PICTURE "@K XXXXXXXXXXXXXXXXXXXXXXXXXXXX" ;
WHEN isediting ;
COLOR ,RGB(0,0,0,255,255,255)
@ 7.385,18.000 SAY "Last Name:" ;
SIZE 0.938,13.500 ;
FONT "Times New Roman", 10 ;
STYLE "BT" ;
PICTURE "@J" ;
COLOR RGB(,,,255,255,255)
@ 7.462,34.333 GET m.last_name ;
SIZE 1.000,28.000 ;
DEFAULT " " ;
FONT "Times New Roman", 8 ;
STYLE "B" ;
PICTURE "@K XXXXXXXXXXXXXXXXXXXXXXXXXXXX" ;
WHEN isediting ;
COLOR ,RGB(0,0,0,255,255,255)
@ 9.231,18.000 SAY "Department:" ;
SIZE 0.938,13.500 ;
FONT "Times New Roman", 10 ;
STYLE "BT" ;
PICTURE "@J" ;

```



COLOR RGB(,,255,255,255)

@ 9.308,34.333 GET m.department ;  
 SIZE 1.000,28.000 ;  
 DEFAULT " " ;  
 FONT "Times New Roman", 8 ;  
 STYLE "B" ;  
 PICTURE "@K XXXXXXXXXXXXXXXXXXXXXXXX" ;  
 WHEN isediting ;  
 COLOR ,RGB(0,0,0,255,255,255)

@ 11.077,18.000 SAY "Nationality:" ;  
 SIZE 0.938,13.500 ;  
 FONT "Times New Roman", 10 ;  
 STYLE "BT" ;  
 PICTURE "@J" ;  
 COLOR RGB(,,255,255,255)

@ 11.154,34.333 GET m.nation ;  
 SIZE 1.000,16.000 ;  
 DEFAULT " " ;  
 FONT "Times New Roman", 8 ;  
 STYLE "B" ;  
 PICTURE "@K XXXXXXXXXXXX" ;  
 WHEN isediting ;  
 COLOR ,RGB(0,0,0,255,255,255)

@ 12.923,18.000 SAY "Class:" ;  
 SIZE 0.938,13.500 ;  
 FONT "Times New Roman", 10 ;  
 STYLE "BT" ;  
 PICTURE "@J" ;  
 COLOR RGB(,,255,255,255)

@ 13.000,34.333 GET m.class ;  
 SIZE 1.000,16.000 ;  
 DEFAULT " " ;  
 FONT "Times New Roman", 8 ;  
 STYLE "B" ;  
 PICTURE "@K XXXXXXXXXXXX" ;  
 WHEN isediting ;  
 COLOR ,RGB(0,0,0,255,255,255)

@ 14.846,0.000 TO 14.846,83.333 ;  
 PEN 2, 8 ;  
 STYLE "1"

@ 16.000,1.667 GET m.top\_btn ;  
 PICTURE "@\*HN \<Top" ;  
 SIZE 1.769,7.833,0.667 ;  
 DEFAULT 1 ;  
 FONT "MS Sans Serif", 8 ;  
 STYLE "B" ;

```

VALID btn_val('TOP') ;
MESSAGE 'Go to first record.'
@ 16.000,9.667 GET m.prev_btn ;
PICTURE "@*HN \<Prev" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('PREV') ;
MESSAGE 'Go to previous record.'
@ 16.000,17.667 GET m.next_btn ;
PICTURE "@*HN \<Next" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('NEXT') ;
MESSAGE 'Go to next record.'
@ 16.000,25.667 GET m.end_btn ;
PICTURE "@*HN \<End" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('END') ;
MESSAGE 'Go to last record.'
@ 16.000,33.667 GET m.loc_btn ;
PICTURE "@*HN \<Locate" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('LOCATE') ;
MESSAGE 'Locate a record.'
@ 16.000,41.667 GET m.add_btn ;
PICTURE "@*HN \<Add" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('ADD') ;
MESSAGE 'Add a new record.'
@ 16.000,49.667 GET m.edit_btn ;
PICTURE "@*HN Ed\<it" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;

```

```

FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('EDIT') ;
MESSAGE 'Edit current record.'
@ 16.000,57.667 GET m.del_btn ;
PICTURE "@*HN \<Delete" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('DELETE') ;
MESSAGE 'Delete current record.'
@ 16.000,65.667 GET m.prnt_btn ;
PICTURE "@*HN P\<rint" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('PRINT') ;
MESSAGE 'Print report.'
@ 16.000,73.667 GET m.exit_btn ;
PICTURE "@*HN \<Close" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('EXIT') ;
MESSAGE 'Close screen.'
@ 0.385,15.667 SAY "STUDENT INFORMATION" ;
FONT "Times New Roman", 18 ;
STYLE "BT"

```

```

IF NOT WVISIBLE("_sb60wp9tq")
    ACTIVATE WINDOW _sb60wp9tq
ENDIF

```

```

* *****
* *
* *   WindowsREAD contains clauses from SCREEN talib
* *
* *****
*

```

```

READ CYCLE ;
    ACTIVATE READACT() ;

```

```

    DEACTIVATE READDEAC() ;
    NOLOCK

RELEASE WINDOW _sb60wp9tq

*      *****
*      *
*      *      Windows Closing Databases
*      *
*      *****
*

IF USED("talib")
    SELECT talib
    USE
ENDIF

SELECT (m.currarea)

#REGION 0

SET READBORDER &rborder

IF m.talkstat = "ON"
    SET TALK ON
ENDIF
IF m.compstat = "ON"
    SET COMPATIBLE ON
ENDIF

*      *****
*      *
*      *      TALIB/Windows Cleanup Code
*      *
*      *****
*

#REGION 1
SET DELETED &wzolddelete
SET FIELDS &wzfields
ON ERROR &wzolderror
ON KEY LABEL ESCAPE &wzoldesc
DO CASE
CASE _DOS AND SET('DISPLAY')='VGA25'

```



```

        @24,0 CLEAR TO 24,79
CASE _DOS AND SET('DISPLAY')='VGA50'
        @49,0 CLEAR TO 49,79
CASE _DOS
        @24,0 CLEAR TO 24,79
ENDCASE

```

\*\*\*\*Procedures\*\*\*\*

```

* *****
*
* * TALIB/Windows Supporting Procedures and Functions
*
* *****
*

```

```

#REGION 1
PROCEDURE readdeac
    IF isediting
        ACTIVATE WINDOW '_sb60wp9tq'
        WAIT WINDOW C_EDITS NOWAIT
    ENDIF
    IF !WVISIBLE(WOUTPUT())
        CLEAR READ
        RETURN .T.
    ENDIF
    RETURN .F.

```

```

PROCEDURE readact
    IF isediting
        SELECT (m.wzalias)
        SHOW GETS
    ENDIF
    DO REFRESH
    RETURN

```

```

PROCEDURE wizerrorhandler
    * This very simple error handler is primarily intended
    * to trap for General field OLE errors which may occur
    * during editing from the MODIFY GENERAL window.
    WAIT WINDOW message()
    RETURN

```

```

PROCEDURE printrec

```

```

PRIVATE sOldError,wizfname,saverec,savearea,tmpcurs,tmpstr
PRIVATE prnt_btn,p_recs,p_output,pr_out,pr_record
STORE 1 TO p_recs,p_output
STORE 0 TO prnt_btn
STORE RECNO() TO saverec
m.sOldError=ON('error')
DO pdialog
IF m.prnt_btn = 2
    RETURN
ENDIF
IF !FILE(ALIAS()+'.FRX')
    m.wizfname=SYS(2004)+'WIZARDS\'+ 'WIZARD.APP'
    IF !FILE(m.wizfname)
        ON ERROR *
        m.wizfname=LOCFILE('WIZARD.APP','APP',C_LOCWIZ)
        ON ERROR &sOldError
        IF !'WIZARD.APP'$UPPER(m.wizfname)
            WAIT WINDOW C_NOWIZ
            RETURN
        ENDIF
    ENDIF
ENDIF
WAIT WINDOW C_MAKEREPO NOWAIT
m.savearea=SELECT()
m.tmpcurs='_'+LEFT(SYS(3),7)
CREATE CURSOR (m.tmpcurs) (comment m)
m.tmpstr = '* LAYOUT = COLUMNAR'+CHR(13)+CHR(10)
INSERT INTO (m.tmpcurs) VALUES(m.tmpstr)
SELECT (m.savearea)
DO (m.wizfname) WITH
", 'WZ_QREPO','NOSCRN/CREATE',ALIAS(),m.tmpcurs
USE IN (m.tmpcurs)
WAIT CLEAR
IF !FILE(ALIAS()+'.FRX') &&wizard could not create report
    WAIT WINDOW C_NOREPO
    RETURN
ENDIF
ENDIF

m.pr_out=IIF(m.p_output=1,'TO PRINT NOCONSOLE','PREVIEW')
m.pr_record=IIF(m.p_recs=1,'NEXT 1','ALL')
REPORT FORM (ALIAS()) &pr_out &pr_record
GO m.saverec

RETURN

```

PROCEDURE BTN\_VAL

```

PARAMETER m.btnname
DO CASE
CASE m.btnname='TOP'
    GO TOP
    WAIT WINDOW C_TOPFILE NOWAIT
CASE m.btnname='PREV'
    IF !BOF()
        SKIP -1
    ENDIF
    IF BOF()
        WAIT WINDOW C_TOPFILE NOWAIT
        GO TOP
    ENDIF
CASE m.btnname='NEXT'
    IF !EOF()
        SKIP 1
    ENDIF
    IF EOF()
        WAIT WINDOW C_ENDFILE NOWAIT
        GO BOTTOM
    ENDIF
CASE m.btnname='END'
    GO BOTTOM
    WAIT WINDOW C_ENDFILE NOWAIT
CASE m.btnname='LOCATE'
    DO loc_dlog
CASE m.btnname='ADD' AND !isediting &&add record
    isediting=.T.
    isadding=.T.
    =edithand('ADD')
    _curobj=1
    DO refresh
    SHOW GETS
    RETURN
CASE m.btnname='EDIT' AND !isediting &&edit record
    IF EOF() OR BOF()
        WAIT WINDOW C_ENDFILE NOWAIT
        RETURN
    ENDIF
    IF RLOCK()
        isediting=.T.
        _curobj=1
        DO refresh
        RETURN
    ELSE
        WAIT WINDOW C_NOLOCK

```

```

ENDIF
CASE m.btnname='EDIT' AND isediting &&save record
IF isadding
    =edithand('SAVE')
ELSE
    GATHER MEMVAR MEMO
ENDIF
UNLOCK
isediting=.F.
isadding=.F.
DO refresh
CASE m.btnname='DELETE' AND isediting &&cancel record
IF isadding
    =edithand('CANCEL')
ENDIF
isediting=.F.
isadding=.F.
UNLOCK
WAIT WINDOW C_ECANCEL NOWAIT
DO refresh
CASE m.btnname='DELETE'
IF EOF() OR BOF()
    WAIT WINDOW C_ENDFILE NOWAIT
    RETURN
ENDIF
IF fox_alert(C_DELREC)
    DELETE
    IF !EOF() AND DELETED()
        SKIP 1
    ENDIF
    IF EOF()
        WAIT WINDOW C_ENDFILE NOWAIT
        GO BOTTOM
    ENDIF
ENDIF
CASE m.btnname='PRINT'
DO printrec
RETURN
CASE m.btnname='EXIT'
m.bailout=.T. &&this is needed if used with FoxApp
CLEAR READ
RETURN
ENDCASE
SCATTER MEMVAR MEMO
SHOW GETS
RETURN

```



## PROCEDURE REFRESH

```
DO CASE
CASE m.isreadonly AND RECCOUNT()=0
    SHOW GETS DISABLE
    SHOW GET exit_btn ENABLE
CASE m.isreadonly
    SHOW GET add_btn DISABLE
    SHOW GET del_btn DISABLE
    SHOW GET edit_btn DISABLE
CASE (RECCOUNT()=0 OR EOF()) AND !m.isediting
    SHOW GETS DISABLE
    SHOW GET add_btn ENABLE
    SHOW GET exit_btn ENABLE
CASE m.isediting
    SHOW GET find_drop DISABLE
    SHOW GET top_btn DISABLE
    SHOW GET prev_btn DISABLE
    SHOW GET loc_btn DISABLE
    SHOW GET next_btn DISABLE
    SHOW GET end_btn DISABLE
    SHOW GET add_btn DISABLE
    SHOW GET prnt_btn DISABLE
    SHOW GET exit_btn DISABLE
    SHOW GET edit_btn,1 PROMPT "\<Save"
    SHOW GET del_btn,1 PROMPT "\<Cancel"
    ON KEY LABEL ESCAPE DO BTN_VAL WITH 'DELETE'
    RETURN
OTHERWISE
    SHOW GET edit_btn,1 PROMPT "Ed\<it"
    SHOW GET del_btn,1 PROMPT "\<Delete"
    SHOW GETS ENABLE
ENDCASE
IF m.is2table
    SHOW GET add_btn DISABLE
ENDIF
ON KEY LABEL ESCAPE
RETURN
```

## PROCEDURE edithand

```
PARAMETER m.paction
* procedure handles edits
DO CASE
CASE m.paction = 'ADD'
```

```

        SCATTER MEMVAR MEMO BLANK
CASE m.paction = 'SAVE'
    INSERT INTO (ALIAS()) FROM MEMVAR
CASE m.paction = 'CANCEL'
    * nothing here
ENDCASE
RETURN

PROCEDURE fox_alert
    PARAMETER wzalrtmess
    PRIVATE alrtbtn
    m.alrtbtn=2
    DEFINE WINDOW _qec1ij2t7 AT 0,0 SIZE 8,50 ;
    FONT "MS Sans Serif",10 STYLE 'B' ;
    FLOAT NOCLOSE NOMINIMIZE DOUBLE TITLE WTITLE()
    MOVE WINDOW _qec1ij2t7 CENTER
    ACTIVATE WINDOW _qec1ij2t7 NOSHOWN
    @ 2,(50-txtwidth(wzalrtmess))/2 SAY wzalrtmess;
    FONT "MS Sans Serif", 10 STYLE "B"
    @ 6,18 GET m.alrtbtn ;
    PICTURE "@*HT \<OK;\?!\<Cancel" ;
    SIZE 1.769,8.667,1.333 ;
    FONT "MS Sans Serif", 8 STYLE "B"
    ACTIVATE WINDOW _qec1ij2t7
    READ CYCLE MODAL
    RELEASE WINDOW _qec1ij2t7
RETURN m.alrtbtn=1
PROCEDURE pdialog
    DEFINE WINDOW _qjn12zbvh ;
    AT 0.000,0.000 ;
    SIZE 13.231,54.800 ;
    TITLE "Microsoft FoxPro" ;
    FONT "MS Sans Serif", 8 ;
    FLOAT NOCLOSE MINIMIZE SYSTEM
    MOVE WINDOW _qjn12zbvh CENTER
    ACTIVATE WINDOW _qjn12zbvh NOSHOWN
    @ 2.846,33.600 SAY "Output:" ;
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
    @ 2.846,4.800 SAY "Print:" ;
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
    @ 4.692,7.200 GET m.p_recs ;
    PICTURE "@*RVN \<Current Record;\<All Records" ;
    SIZE 1.308,18.500,0.308 ;
    DEFAULT 1 ;

```

```

    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
@ 4.692,36.000 GET m.p_output ;
    PICTURE "@*RVN \<Printer,Pre\<view" ;
    SIZE 1.308,12.000,0.308 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
@ 10.154,16.600 GET m.prnt_btn ;
    PICTURE "@*HT P\<rint;Ca\<ncel" ;
    SIZE 1.769,8.667,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B"
    ACTIVATE WINDOW _qjn12zbvh
    READ CYCLE MODAL
    RELEASE WINDOW _qjn12zbvh
RETURN
PROCEDURE loc_dlog
    PRIVATE gfields,i
    DEFINE WINDOW wzlocate FROM 1,1 TO 20,40;
        SYSTEM GROW CLOSE ZOOM FLOAT FONT "MS Sans Serif",8
    MOVE WINDOW wzlocate CENTER
    m.gfields=SET('FIELDS',2)
    IF !EMPTY(RELATION(1))
        SET FIELDS ON
        IF m.gfields # 'GLOBAL'
            SET FIELDS GLOBAL
        ENDIF
        IF EMPTY(FLDLIST())
            m.i=1
            DO WHILE !EMPTY(OBJVAR(m.i))
                IF ATC('M.',OBJVAR(m.i))=0
                    SET FIELDS TO (OBJVAR(m.i))
                ENDIF
                m.i = m.i + 1
            ENDDO
        ENDIF
    ENDIF
    BROWSE WINDOW wzlocate NOEDIT NODELETE ;
        NOMENU TITLE C_BRTITLE
    SET FIELDS &gfields
    SET FIELDS OFF
    RELEASE WINDOW wzlocate
RETURN

```

## 8. Borrow A Book

This screen is used for borrowing a book and calculating charge if there is. Here we use some mathematical calculation to calculate the Available Books, Expect Return Date, Total Days, Late Days and charges. The calculations are shown below:

### FUNCTION AVAILABLE

$m.avilable = m.no\_of\_copy - m.borrowed$   
RETURN

### FUNCTION EXPECT RETURN DATE

$m.exp\_date = m.issue\_date + 30$   
RETURN

### FUNCTION TOTALDAYS

$m.total\_days = m.ret\_date - m.issue\_date$

### FUNCTION LATEDAYS

IF  $m.total\_days \leq 30$  THEN  
   $m.late\_days = 0$   
ELSE  
   $m.late\_days = m.total\_days - 30$   
ENDIF

### FUNCTION CHARGE

$m.charge = m.late\_days * 0.5$

Screen shown as 8. *Borrow A Book*



LIBRARY MENU

GENERAL MENU

Dain

## BORROW & CHARGES

Student Id:	960147	
Serial No:	25	
Book Name:	VISUAL BASIC	
No_of_Copy:	3	
Borrowed:	1	
Avilable:	2	
Issue Date:	12.01.2000	Expect Ret Date: 11.02.2000
Return Date:	24.01.2000	Total Days: 12
Late Days:	0	Charge: 0,0 \$

Top Prev Next End Locate Add Edit Delete Print Close

Another thing our program can do is to INFORM students who's borrowed the book and they didn't bring back. For this option when the employee is in charge by pressing the buttons Next or Preview he/she can understand whose those. The screen for that reason is shown below which contains the Student Number, Book Name and Issue Date.

```
The person did not bring the book back yet,who has ;  
ST_ID      : 980545  
BOOK NAME  : BUILDING STRUCTURE  
ISSUE DATE : 21/1/1998
```

One more option we can get in this program that, each student can get just one book can borrow not more. Here when the student number is entered and all the variables will be asked for borrowed a book. In the normal situation we have to save it, but here it can't save it and it will give the following screen which includes the message and information about last book that borrowed.

```
The person did not bring the book back yet,who has ;  
ST_ID      : 960577  
BOOK NAME  : DATA STRUCTURE  
ISSUE DATE : 1/12/2000  
This person can not borrow a book because borrowed one before!
```

**Program of how to create borrow screen and the operations are going on:**

```

* *****
*
* * 21.01.00          DAIN.SPR          15:30:06
* *
* *****
*
* * Author's Name
* *
* * Copyright (c) 2000 Company Name
* * Address
* * City,   Zip
* *
* * Description:
* * This program was automatically generated by GENSCRN.
* *
* *****

```

```

* *****
*
* *          DAIN/Windows Setup Code - SECTION 1
* *
* *****
*

```

```

#REGION 1
PRIVATE wzfields,wztalk
IF SET("TALK") = "ON"
    SET TALK OFF
    m.wztalk = "ON"
ELSE
    m.wztalk = "OFF"
ENDIF
m.wzfields=SET('FIELDS')
SET FIELDS OFF
IF m.wztalk = "ON"
    SET TALK ON
ENDIF

```

```

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"

```



```

        SET TALK OFF
        m.talkstat = "ON"
ELSE
        m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.rborder = SET("READBORDER")
SET READBORDER ON

m.currarea = SELECT()

```

```

* *****
*
* *   S5806443/Windows Databases, Indexes, Relations
* *
* *****
*

```

```

IF USED("dain")
    SELECT dain
    SET ORDER TO TAG "_sb60wvu46"
ELSE
    SELECT 0
    USE (LOCFILE("dain.dbf","DBF","Where is dain?"));
    AGAIN ALIAS dain ;
    ORDER TAG "_sb60wvu46"
ENDIF

```

```

* *****
*
* *   Windows Window definitions
* *
* *****
*

```

```

IF NOT WEXIST("_0010x84j1")
    DEFINE WINDOW _0010x84j1 ;
        AT 0.000,0.000 ;
        SIZE 21.538,85.167 ;
        TITLE "Dain" ;
        FONT "MS Sans Serif", 8 ;
        STYLE "B" ;
        FLOAT ;
        CLOSE ;
        MINIMIZE ;
        COLOR RGB(,,192,192,192)

```



```

        MOVE WINDOW _0010x84j1 CENTER
    ENDIF

```

```

*      *****
*
*      *      DAIN/Windows Setup Code - SECTION 2
*      *
*      *****
*

```

```

#REGION 1

```

```

#DEFINE C_DBFEMPTY      'Database is empty, add a record?'
#DEFINE C_EDITS          'Please finish your edits.'
#DEFINE C_TOPFILE       'Top of file.'
#DEFINE C_ENDFILE       'End of file.'
#DEFINE C_BRTITLE       'Locate Record'
#DEFINE C_NOLOCK        'Sorry, could not lock record -- try again later.'
#DEFINE C_ECANCEL       'Edits Canceled.'
#DEFINE C_DELREC        'Delete selected record?'
#DEFINE C_NOFEAT        'Feature not available yet.'
#DEFINE C_NOWIZ         'Wizard application is not available.'
#DEFINE C_MAKEREPO      'Creating report with Report Wizard.'
#DEFINE C_NOREPO        'Could not create report.'
#DEFINE C_DELNOTE       'Deleting records...'
#DEFINE C_READONLY      'Table is read-only. No editing allowed.'
#DEFINE C_NOTABLE       'No table selected. Open table or run query.'
#DEFINE C_BADEXP        'Invalid expression.'
#DEFINE C_LOCWIZ        'Locate WIZARD.APP.'
#DEFINE C_MULTITABLE    'You have multiple related tables. Adding records in not
allowed.'
#DEFINE C_C             'This person didn't bring the book back in the time'

```

```

MOVE WINDOW '_0010x84j1' CENTER
PRIVATE isediting,isadding,wztblarr
PRIVATE wzolddelete,wzolderror,wzoldesc
PRIVATE wzalias, tempcurs,wzlastrec
PRIVATE isreadonly,find_drop,is2table

```

```

IF EMPTY(ALIAS())
    WAIT WINDOW C_NOTABLE
    RETURN
ENDIF

```

```

m.wztblarr= "
m.wzalias=SELECT()
m.isediting=.F.
m.isadding=.F.
m.is2table = .F.

```

```

m.wzolddelete=SET('DELETE')
SET DELETED ON
m.tempcurs=SYS(2015) &&used if General field
m.wzlastrec = 1
m.wzolderror=ON('error')
ON ERROR DO wizerrorhandler
wzoldesc=ON('KEY','ESCAPE')
ON KEY LABEL ESCAPE
m.find_drop = IIF(_DOS,0,2)

m.isreadonly=IIF(ISREAD(),.T.,.F.)
IF m.isreadonly
    WAIT WINDOW C_READONLY TIMEOUT 1
ENDIF

```

```

IF RECCOUNT()=0 AND !m.isreadonly AND fox_alert(C_DBFEMPTY)
    APPEND BLANK
ENDIF

```

```

GOTO TOP
SCATTER MEMVAR MEMO

```

```

* *****
* *
* *      DAIN/Windows Screen Layout
* *
* *****
*

```

```

#REGION 1
IF WVISIBLE("_0010x84j1")
    ACTIVATE WINDOW _0010x84j1 SAME
ELSE
    ACTIVATE WINDOW _0010x84j1 NOSHOW
ENDIF
@ 2.846,0.833 TO 2.846,84.166 ;
    PEN 2, 8 ;
    STYLE "1"
@ 18.077,0.833 TO 18.077,84.166 ;
    PEN 2, 8 ;
    STYLE "1"
@ 0.385,20.500 SAY "BORROW & CHARGES" ;
    FONT "Times New Roman", 18 ;
    STYLE "BT"
@ 16.538,80.833 SAY "$" ;
    FONT "MS Sans Serif", 10 ;
    STYLE "BT"
@ 3.385,12.333 SAY "Student Id:" ;
    SIZE 0.938,18.800 ;

```

```

FONT "Times New Roman", 9 ;
STYLE "BT" ;
PICTURE "@J" ;
COLOR RGB(,,255,255,255)
@ 3.462,30.167 GET m.st_id ;
SIZE 1.000,15.600 ;
DEFAULT " " ;
FONT "Times New Roman", 8 ;
STYLE "B" ;
PICTURE "@K 9999999999" ;
WHEN isediting ;
COLOR ,RGB(0,0,0,255,255,255)
@ 5.000,12.333 SAY "Serial No:" ;
SIZE 0.938,18.800 ;
FONT "Times New Roman", 9 ;
STYLE "BT" ;
PICTURE "@J" ;
COLOR RGB(,,255,255,255)
@ 5.077,30.167 GET m.serial_no ;
SIZE 1.000,15.600 ;
DEFAULT " " ;
FONT "Times New Roman", 8 ;
STYLE "B" ;
PICTURE "@K 9999999999" ;
WHEN isediting ;
COLOR ,RGB(0,0,0,255,255,255)
@ 6.692,12.333 SAY "Book Name:" ;
SIZE 0.938,18.800 ;
FONT "Times New Roman", 9 ;
STYLE "BT" ;
PICTURE "@J" ;
COLOR RGB(,,255,255,255)
@ 6.769,30.167 GET m.book_name ;
SIZE 1.000,51.800 ;
DEFAULT " " ;
FONT "Times New Roman", 8 ;
STYLE "B" ;
PICTURE "@K XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" ;
WHEN ROSY() ;
COLOR ,RGB(0,0,0,255,255,255)
@ 8.308,12.333 SAY "No_of_Copy:" ;
SIZE 0.938,18.800 ;
FONT "Times New Roman", 9 ;
STYLE "BT" ;
PICTURE "@J" ;
COLOR RGB(,,255,255,255)
@ 8.385,30.333 GET m.No_of_Copy ;
SIZE 1.000,3.600 ;
DEFAULT " " ;
FONT "Times New Roman", 8 ;

```



STYLE "B" ;  
 PICTURE "@K 999" ;  
 WHEN ROSY1() ;  
 COLOR ,RGB(0,0,0,255,255,255)  
 @ 9.923,12.333 SAY "Borrowed:" ;  
 SIZE 0.938,18.800 ;  
 FONT "Times New Roman", 9 ;  
 STYLE "BT" ;  
 PICTURE "@J" ;  
 COLOR RGB(,,255,255,255)  
 @ 10.000,30.333 GET m.borrowed ;  
 SIZE 1.000,3.600 ;  
 DEFAULT " " ;  
 FONT "Times New Roman", 8 ;  
 STYLE "B" ;  
 PICTURE "@K 999" ;  
 WHEN isediting ;  
 COLOR ,RGB(0,0,0,255,255,255)  
 @ 11.538,12.333 SAY "Avilable:" ;  
 SIZE 0.938,18.800 ;  
 FONT "Times New Roman", 9 ;  
 STYLE "BT" ;  
 PICTURE "@J" ;  
 COLOR RGB(,,255,255,255)  
 @ 11.615,30.333 GET m.avilable ;  
 SIZE 1.000,3.600 ;  
 DEFAULT " " ;  
 FONT "Times New Roman", 8 ;  
 STYLE "B" ;  
 PICTURE "@K 999" ;  
 WHEN AA() ;  
 COLOR ,RGB(0,0,0,255,255,255)  
 @ 13.077,12.333 SAY "Issue Date:" ;  
 SIZE 0.938,18.800 ;  
 FONT "Times New Roman", 9 ;  
 STYLE "BT" ;  
 PICTURE "@J" ;  
 COLOR RGB(,,255,255,255)  
 @ 13.154,30.167 GET m.issue\_date ;  
 SIZE 1.000,16.000 ;  
 DEFAULT " " ;  
 FONT "Times New Roman", 8 ;  
 STYLE "B" ;  
 PICTURE "@K XXXXXXXXXXXX" ;  
 WHEN isediting ;  
 COLOR ,RGB(0,0,0,255,255,255)  
 @ 13.154,48.333 SAY "Expect Ret Date:" ;  
 SIZE 0.938,18.800 ;  
 FONT "Times New Roman", 9 ;  
 STYLE "BT" ;



COLOR RGB(,,255,255,255)  
 @ 13.154,66.167 GET m.exp\_date ;  
 SIZE 1.000,16.000 ;  
 DEFAULT { / / } ;  
 FONT "Times New Roman", 8 ;  
 STYLE "B" ;  
 PICTURE "@K XXXXXXXXXXXX" ;  
 WHEN WW() ;  
 COLOR ,RGB(0,0,0,255,255,255)  
 @ 14.846,12.333 SAY "Return Date:" ;  
 SIZE 0.938,18.800 ;  
 FONT "Times New Roman", 9 ;  
 STYLE "BT" ;  
 PICTURE "@J" ;  
 COLOR RGB(,,255,255,255)  
 @ 14.846,48.333 SAY "Total Days:" ;  
 SIZE 0.938,18.800 ;  
 FONT "Times New Roman", 9 ;  
 STYLE "BT" ;  
 PICTURE "@J" ;  
 COLOR RGB(,,255,255,255)  
 @ 14.923,30.167 GET m.ret\_date ;  
 SIZE 1.000,16.000 ;  
 DEFAULT " " ;  
 FONT "Times New Roman", 8 ;  
 STYLE "B" ;  
 PICTURE "@K XXXXXXXXXXXX" ;  
 WHEN isediting ;  
 COLOR ,RGB(0,0,0,255,255,255)  
 @ 14.923,66.333 GET m.total\_days ;  
 SIZE 1.000,3.600 ;  
 DEFAULT " " ;  
 FONT "Times New Roman", 8 ;  
 STYLE "B" ;  
 PICTURE "@K 999" ;  
 WHEN TOTALDAYS() ;  
 COLOR ,RGB(0,0,0,255,255,255)  
 @ 16.538,12.333 SAY "Late Days:" ;  
 SIZE 0.938,18.800 ;  
 FONT "Times New Roman", 9 ;  
 STYLE "BT" ;  
 PICTURE "@J" ;  
 COLOR RGB(,,255,255,255)  
 @ 16.538,48.500 SAY "Charge:" ;  
 SIZE 0.938,18.800 ;  
 FONT "Times New Roman", 9 ;  
 STYLE "BT" ;  
 PICTURE "@J" ;  
 COLOR RGB(,,255,255,255)  
 @ 16.615,30.500 GET m.late\_days ;

```

SIZE 1.000,3.600 ;
DEFAULT " " ;
FONT "Times New Roman", 8 ;
STYLE "B" ;
PICTURE "@K 999" ;
WHEN LATEDAYS() ;
COLOR ,RGB(0,0,0,255,255,255)
@ 16.615,66.333 GET m.charge ;
SIZE 1.000,15.600 ;
DEFAULT " " ;
FONT "Times New Roman", 8 ;
STYLE "B" ;
PICTURE "@K 99.9" ;
WHEN CHARGE() ;
COLOR ,RGB(0,0,0,255,255,255)
@ 18.769,2.500 GET m.top_btn ;
PICTURE "@*HN \<Top" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('TOP') ;
MESSAGE 'Go to first record.'
@ 18.769,10.500 GET m.prev_btn ;
PICTURE "@*HN \<Prev" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('PREV') ;
MESSAGE 'Go to previous record.'
@ 18.769,18.500 GET m.next_btn ;
PICTURE "@*HN \<Next" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('NEXT') ;
MESSAGE 'Go to next record.'
@ 18.769,26.500 GET m.end_btn ;
PICTURE "@*HN \<End" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('END') ;
MESSAGE 'Go to last record.'
@ 18.769,34.500 GET m.loc_btn ;
PICTURE "@*HN \<Locate" ;
SIZE 1.769,7.833,0.667 ;

```

```

DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('LOCATE') ;
MESSAGE 'Locate a record.'
@ 18.769,42.500 GET m.add_btn ;
PICTURE "@*HN \<Add" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('ADD') ;
MESSAGE 'Add a new record.'
@ 18.769,50.500 GET m.edit_btn ;
PICTURE "@*HN Ed\<it" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('EDIT') ;
MESSAGE 'Edit current record.'
@ 18.769,58.500 GET m.del_btn ;
PICTURE "@*HN \<Delete" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('DELETE') ;
MESSAGE 'Delete current record.'
@ 18.769,66.500 GET m.prnt_btn ;
PICTURE "@*HN P\<rint" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('PRINT') ;
MESSAGE 'Print report.'
@ 18.769,74.500 GET m.exit_btn ;
PICTURE "@*HN \<Close" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('EXIT') ;
MESSAGE 'Close screen.'

```

```

IF NOT WVISIBLE("_0010x84j1")
    ACTIVATE WINDOW _0010x84j1
ENDIF

```



```

*****
*
*   WindowsREAD contains clauses from SCREEN s5806443
*
*****

```

```

READ CYCLE ;
    ACTIVATE READACT() ;
    DEACTIVATE READDEAC() ;
    NOLOCK

```

```

RELEASE WINDOW _0010x84j1

```

```

*   *****
*   *
*   *       Windows Closing Databases
*   *
*   *****
*

```

```

IF USED("dain")
    SELECT dain
    USE
ENDIF

```

```

SELECT (m.currarea)

```

```

#REGION 0

```

```

SET READBORDER &rborder

```

```

IF m.talkstat = "ON"
    SET TALK ON

```

```

ENDIF

```

```

IF m.compstat = "ON"
    SET COMPATIBLE ON

```

```

ENDIF

```

```

*   *****
*   *
*   *       DAIN/Windows Cleanup Code
*   *
*   *****
*

```

```

#REGION 1

```



```

SET DELETED &wzolddelete
SET FIELDS &wzfields
ON ERROR &wzolderror
ON KEY LABEL ESCAPE &wzoldesc
DO CASE
CASE _DOS AND SET('DISPLAY')='VGA25'
    @24,0 CLEAR TO 24,79
CASE _DOS AND SET('DISPLAY')='VGA50'
    @49,0 CLEAR TO 49,79
CASE _DOS
    @24,0 CLEAR TO 24,79
ENDCASE

```

\*\*\*\*Procedures\*\*\*\*

```

* *****
* *
* *   DAIN/Windows Supporting Procedures and Functions
* *
* *****
*

```

#REGION 1

PROCEDURE readdeac

```

IF isediting
    ACTIVATE WINDOW '_0010x84j1'
    WAIT WINDOW C_EDITS NOWAIT
ENDIF
IF !WVISIBLE(WOUTPUT())
    CLEAR READ
    RETURN .T.
ENDIF
RETURN .F.

```

PROCEDURE readact

```

IF !isediting
    SELECT (m.wzalias)
    SHOW GETS
ENDIF
DO REFRESH
RETURN

```

PROCEDURE wizerrorhandler

```

* This very simple error handler is primarily intended
* to trap for General field OLE errors which may occur
* during editing from the MODIFY GENERAL window.
    WAIT WINDOW message()
RETURN

```

```

PROCEDURE printrec
PRIVATE sOldError,wizfname,saverec,savearea,tmpcurs,tmpstr
PRIVATE prnt_btn,p_recs,p_output,pr_out,pr_record
STORE 1 TO p_recs,p_output
STORE 0 TO prnt_btn
STORE RECNO() TO saverec
m.sOldError=ON('error')
DO pdialog
IF m.prnt_btn = 2
RETURN
ENDIF
IF !FILE(ALIAS()+'.FRX')
m.wizfname=SYS(2004)+'WIZARDS\'+ 'WIZARD.APP'
IF !FILE(m.wizfname)
ON ERROR *
m.wizfname=LOCFILE('WIZARD.APP','APP',C_LOCWIZ)
ON ERROR &sOldError
IF !'WIZARD.APP'$UPPER(m.wizfname)
WAIT WINDOW C_NOWIZ
RETURN
ENDIF
ENDIF
WAIT WINDOW C_MAKEREPO NOWAIT
m.savearea=SELECT()
m.tmpcurs='_'+LEFT(SYS(3),7)
CREATE CURSOR (m.tmpcurs) (comment m)
m.tmpstr = '* LAYOUT = COLUMNAR'+CHR(13)+CHR(10)
INSERT INTO (m.tmpcurs) VALUES(m.tmpstr)
SELECT (m.savearea)
DO (m.wizfname) WITH
", 'WZ_QREPO', 'NOSCRN/CREATE', ALIAS(), m.tmpcurs
USE IN (m.tmpcurs)
WAIT CLEAR
IF !FILE(ALIAS()+'.FRX') &&wizard could not create report
WAIT WINDOW C_NOREPO
RETURN
ENDIF
ENDIF

m.pr_out=IIF(m.p_output=1,'TO PRINT NOCONSOLE','PREVIEW')
m.pr_record=IIF(m.p_recs=1,'NEXT 1','ALL')
REPORT FORM (ALIAS()) &pr_out &pr_record
GO m.saverec

RETURN

```

```

PROCEDURE BTN_VAL
PARAMETER m.btnname
DO CASE

```

```

CASE m.btnname='TOP'
    GO TOP
    WAIT WINDOW C_TOPFILE NOWAIT
CASE m.btnname='PREV'
    IF !BOF()
        SKIP -1
    ENDIF
    IF BOF()
        WAIT WINDOW C_TOPFILE NOWAIT
        GO TOP
    ENDIF
CASE m.btnname='NEXT'
    IF !EOF()
        SKIP 1
    ENDIF
    IF EOF()
        WAIT WINDOW C_ENDFILE NOWAIT
        GO BOTTOM
    ENDIF
CASE m.btnname='END'
    GO BOTTOM
    WAIT WINDOW C_ENDFILE NOWAIT
CASE m.btnname='LOCATE'
    DO loc_dlog
CASE m.btnname='ADD' AND !isediting &&add record
    isediting=.T.
    isadding=.T.
    =edithand('ADD')
    _curobj=1
    DO refresh
    SHOW GETS
    RETURN
CASE m.btnname='EDIT' AND !isediting &&edit record
    IF EOF() OR BOF()
        WAIT WINDOW C_ENDFILE NOWAIT
        RETURN
    ENDIF
    IF RLOCK()
        isediting=.T.
        _curobj=1
        DO refresh
        RETURN
    ELSE
        WAIT WINDOW C_NOLOCK
    ENDIF
CASE m.btnname='EDIT' AND isediting &&save record
    IF isadding
        =edithand('SAVE')
    ELSE
        GATHER MEMVAR MEMO

```



```

ENDIF
UNLOCK
isediting=.F.
isadding=.F.
DO refresh
CASE m.btnname='DELETE' AND isediting      &&cancel record
  IF isadding =edithand('CANCEL')
  ENDIF
  isediting=.F.
  isadding=.F.
  UNLOCK
  WAIT WINDOW C_ECANCEL NOWAIT
  DO refresh
CASE m.btnname='DELETE'
  IF EOF() OR BOF()
    WAIT WINDOW C_ENDFILE NOWAIT
    RETURN
  ENDIF
  IF fox_alert(C_DELREC)
    DELETE
    IF !EOF() AND DELETED()
      SKIP 1
    ENDIF
    IF EOF()
      WAIT WINDOW C_ENDFILE NOWAIT
      GO BOTTOM
    ENDIF
  ENDIF
ENDIF

CASE m.btnname='PRINT'
  DO printrec
  RETURN
CASE m.btnname='EXIT'
  m.bailout=.T. &&this is needed if used with FoxApp
  CLEAR READ
  RETURN

ENDCASE
SCATTER MEMVAR MEMO

```

### Codes of INFORM students:

```

SHOW GETS
IF (m.issue_date<date()-30)
  WAIT WINDOW "The person did not bring the book back
yet,who has ;"+chr(13);
  +" ST_ID      : "+alltrim(str(m.st_id))+chr(13);
  +" BOOK NAME : "+m.book_name+chr(13);
  +" ISSUE DATE :
"+alltrim(str(day(issue_date)))+"/"+alltrim(str(month(issue_date)))+"/"+alltrim(str(ye
ar(issue_date)))

```



ENDIF

RETURN

PROCEDURE REFRESH

DO CASE

CASE m.isreadonly AND RECCOUNT()=0

SHOW GETS DISABLE

SHOW GET exit\_btn ENABLE

CASE m.isreadonly

SHOW GET add\_btn DISABLE

SHOW GET del\_btn DISABLE

SHOW GET edit\_btn DISABLE

CASE (RECCOUNT()=0 OR EOF()) AND !m.isediting

SHOW GETS DISABLE

SHOW GET add\_btn ENABLE

SHOW GET exit\_btn ENABLE

CASE m.isediting

SHOW GET find\_drop DISABLE

SHOW GET top\_btn DISABLE

SHOW GET prev\_btn DISABLE

SHOW GET loc\_btn DISABLE

SHOW GET next\_btn DISABLE

SHOW GET end\_btn DISABLE

SHOW GET add\_btn DISABLE

SHOW GET prnt\_btn DISABLE

SHOW GET exit\_btn DISABLE

SHOW GET edit\_btn,1 PROMPT "\<Save"

SHOW GET del\_btn,1 PROMPT "\<Delete"

ON KEY LABEL ESCAPE DO BTN\_VAL WITH 'DELETE'

RETURN

OTHERWISE

SHOW GET edit\_btn,1 PROMPT "Ed\<it"

SHOW GET del\_btn,1 PROMPT "\<Delete"

SHOW GETS ENABLE

ENDCASE

IF m.is2table

SHOW GET add\_btn DISABLE

ENDIF

ON KEY LABEL ESCAPE

RETURN

PROCEDURE edithand

PARAMETER m.paction

\*procedure handles edits

## Codes of checking for student who's borrowed book before

DO CASE

CASE m.paction = 'ADD'

SCATTER MEMVAR MEMO BLANK

CASE m.paction = 'SAVE'

ck1="True"

value1=m.st\_id

value2=recno()

for c=1 to RecCount()

go c

IF ((m.issue\_date<date()-30)and(value1=st\_id))

WAIT WINDOW "The person did not bring the book back

yet,who has ;"+chr(13);

+" ST\_ID : "+alltrim(str(st\_id))+chr(13);

+" BOOK NAME : "+book\_name+chr(13);

+" ISSUE DATE :

"+alltrim(str(day(issue\_date)))+"/"+alltrim(str(month(issue\_date)))+"/"+alltrim(str(year(issue\_date)));

+chr(13)+"This person can not borrow a book

because borrowed one before!"

ck1="False"

ENDIF

endfor

if ck1="True"

INSERT INTO (ALIAS()) FROM MEMVAR

endif

go value2

CASE m.paction = 'CANCEL'

\* nothing here

ENDCASE

RETURN

PROCEDURE fox\_alert

PARAMETER wzalrtmess

PRIVATE alrtbtn

m.alrtbtn=2

DEFINE WINDOW \_qec1ij2t7 AT 0,0 SIZE 8,50 ;

FONT "MS Sans Serif",10 STYLE 'B' ;

FLOAT NOCLOSE NOMINIMIZE DOUBLE TITLE WTITLE()

MOVE WINDOW \_qec1ij2t7 CENTER

ACTIVATE WINDOW \_qec1ij2t7 NOSHOW

@ 2,(50-txtwidth(wzalrtmess))/2 SAY wzalrtmess;

FONT "MS Sans Serif", 10 STYLE "B"

@ 6,18 GET m.alrtbtn ;

PICTURE "@\*HT \<OK;\?!<Cancel" ;

SIZE 1.769,8.667,1.333 ;

FONT "MS Sans Serif", 8 STYLE "B"

ACTIVATE WINDOW \_qec1ij2t7

```
READ CYCLE MODAL
RELEASE WINDOW _qec1ij2t7
RETURN m.alrtbtn=1
```

```
PROCEDURE pdialog
  DEFINE WINDOW _qjn12zbvh ;
    AT 0.000,0.000 ;
    SIZE 13.231,54.800 ;
    TITLE "Microsoft FoxPro" ;
    FONT "MS Sans Serif", 8 ;
    FLOAT NOCLOSE MINIMIZE SYSTEM
  MOVE WINDOW _qjn12zbvh CENTER
  ACTIVATE WINDOW _qjn12zbvh NOSHOW
  @ 2.846,33.600 SAY "Output:" ;
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
  @ 2.846,4.800 SAY "Print:" ;
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
  @ 4.692,7.200 GET m.p_recs ;
    PICTURE "@*RVN \<Current Record;\<All Records" ;
    SIZE 1.308,18.500,0.308 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
  @ 4.692,36.000 GET m.p_output ;
    PICTURE "@*RVN \<Printer;Pre\<view" ;
    SIZE 1.308,12.000,0.308 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
  @ 10.154,16.600 GET m.prnt_btn ;
    PICTURE "@*HT P\<rint;Ca\<ncel" ;
    SIZE 1.769,8.667,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B"
  ACTIVATE WINDOW _qjn12zbvh
  READ CYCLE MODAL
  RELEASE WINDOW _qjn12zbvh
RETURN
```

```
PROCEDURE loc_dlog
  PRIVATE gfields,i
  DEFINE WINDOW wzlocate FROM 1,1 TO 20,40;
    SYSTEM GROW CLOSE ZOOM FLOAT FONT "MS Sans Serif",8
  MOVE WINDOW wzlocate CENTER
  m.gfields=SET('FIELDS',2)
```



```

IF !EMPTY(RELATION(1))
    SET FIELDS ON
    IF m.gfields # 'GLOBAL'
        SET FIELDS GLOBAL
    ENDIF
    IF EMPTY(FLDLIST())
        m.i=1
        DO WHILE !EMPTY(OBJVAR(m.i))
            IF ATC('M.',OBJVAR(m.i))=0
                SET FIELDS TO (OBJVAR(m.i))
            ENDIF
            m.i = m.i + 1
        ENDDO
    ENDIF
ENDIF
BROWSE WINDOW wzlocate NOEDIT NODELETE ;
    NOMENU TITLE C_BRTITLE
SET FIELDS &gfields
SET FIELDS OFF
RELEASE WINDOW wzlocate
RETURN

```

```

FUNCTION TOTALDAYS
m.total_days=m.ret_date-m.issue_date

```

```

FUNCTION WW
m.exp_date=m.issue_date+30
RETURN

```

```

FUNCTION LATEDAYS
IF m.total_days<=30 THEN
    m.late_days=0
ELSE
    m.late_days=m.total_days-30
ENDIF

```

```

FUNCTION CHARGE
m.charge=m.late_days*0.5

```

```

FUNCTION ROSY
CLOSE DATABASES
USE (LOCFILE("kitab.dbf","DBF","Where is kitab?"));
    SHARE;
    AGAIN ALIAS kitab ;
    ORDER TAG "so"
SEEK m.serial_no
IF FOUND()
    m.book_name=book_name
ENDIF
show gets

```



```
CLOSE DATABASES
USE (LOCFILE("dain.dbf","dbf","WHERE IS dain ?"));
share ;
AGAIN ALIAS dain ;
ORDER TAG "sno"
```

```
FUNCTION ROSY1
CLOSE DATABASES
USE (LOCFILE("kitab.dbf","DBF","Where is kitab?"));
SHARE;
AGAIN ALIAS kitab ;
ORDER TAG "so"
```

```
SEEK m.serial_no
IF FOUND()
m.No_of_Copy=No_of_copy
ENDIF
show gets
CLOSE DATABASES
USE (LOCFILE("dain.dbf","dbf","WHERE IS dain ?"));
share ;
AGAIN ALIAS dain ;
ORDER TAG "sno"
```

```
FUNCTION AA
m.avilable=m.no_of_copy-m.borrowed
RETURN
```

## CONCLUSION

This program was designed according to the basic needs of a circulation desk of any library. It has the advantage of fast search ability according to different parameters.

The program is user-friendly even for those people that are not so familiar to the computers. Consequently in a very short training period the library staff could use it easily.

For big libraries a small network could be set up, so simultaneously more than one staff respond to the members for borrowing and similar process. This program could be easily adopted for the network use.

More and more options could also be added to the main menu here. For instant options for Periodicals, Scientific Journals, Microfilms and CD's.

One of the problems that I was faced was not to be familiar to the bar code programming. This kind of program should actually support the bar code machines so full automation could be reached in the library.

Generally I could say that I could reach my goals after doing this project as now I have a general view about the programs that create databases.