2014

informatics

PROJECT TITLE: Library Membership Control System

STUDENT NAME: Myo Thet Tun

STUDENT ID: 018800021524

CLASS CODE: P12May2014IT202A

LECTURER NAME: Tng Choon Hock

SUBJECT CODE: IT202

SUBJECT: C Programming

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Library's background

Knowledge Planet Library is a library of "Knowledge Planet Academy", a private learning center founded in the year 2004 and is located in Yangon, Myanmar. The library was officially opened in the year 2005 for the purpose of getting more references, knowledge and information for the students of this school. The library was rather a small one when it was founded. Now, over the past 9 years, the size of the library is getting bigger because the library's officials have been buying many new kinds of books to become a better library and the students' knowledge center point. And the increasing number of members makes the staffs difficult to handle the records manually.

1.2 Current system

Since the library was founded, the staffs have been doing all the works manually. That is, the librarians have to record everyday transactions manually on the paper- based system. He or she has to write down every detail in the books daily. It's really a time consuming task and data inconsistency is also very high. Below are the problems that current system is facing with.

1.3 <u>Problems background</u>

Slow Retrieval of Data - The information is stored in different parts of the site and it may take a long time to retrieve or update the data. Sometimes, it can take up to 20 or 30 minutes finding the relevant information.

Paper Wastage - Much paper is waste due to the number of records daily and number of library members. Duplication of data can be occurred by repeating the same thing over and over.

Unproductive use of storage space - Paper takes up a massive amount of room in the site.

Poor Customer Service - Sometimes, the information needed may be unavailable.

No reliable database system - The records on paper can be lost or damaged at any time. Since there is no backup for the data, the lost or damaged documents cannot be regained at all.

No security system – Writing on paper is totally lack of security system. Any unauthorized person may view, update or even the data can be stolen.

These are the weak points that the current system is facing with.

1.3 <u>**Objectives**</u>

The main purpose of the new system is to handle all the problems that the old system is currently facing with. With the newly implemented system, the records can be easily created. There will be no duplication of data records because all the data records are controlled by a record key called ID. There is also an error checking method to detect the data type error (e.g. the record cannot be stored if the user types the alphabet character where a numeric value must be typed in). For retrieval of the records, it is much faster and easier than the previous system. Editing and deleting also can be easily and quickly performed. The records are to be stored in the security enabled database system which can store many lines of data records. There will be no store room at all to store the documents and no paper wastage at all. So, all the current problems will be solved with this single "Library Membership Control System". The staff will only need to sit at the workstation just to perform all these tasks.

These are the advantages of the new "Library Membership Control System".

1.5 Record Database

Library Membership Control System

Main Menu

1. Add New Member Record

2. Display Member Record

3. Search Member Record

4. Update Member Record

5. Delete Member Record

6. Exit

1. Add New Member Record

- ❖ Enter Member ID
- ❖ Enter Name
- ❖ Enter IC No.
- ❖ Enter Gender
- ❖ Enter Phone
- ❖ Enter Address
- ❖ Enter Registration Date
- ❖ Enter Expiry Date

2. Display Member Record

- ❖ Display Member ID
- ❖ Display Name
- ❖ Display IC No.

❖ Display Gender

❖ Display Phone

❖ Display Address

❖ Display Registration Date

❖ Display Expiry Date

3. Search Member Record

❖ Search record using Member ID

4. Update Member Record

❖ Update Name

❖ Update IC No.

❖ Update Gender

❖ Update Phone

❖ Update Address

❖ Update Registration Date
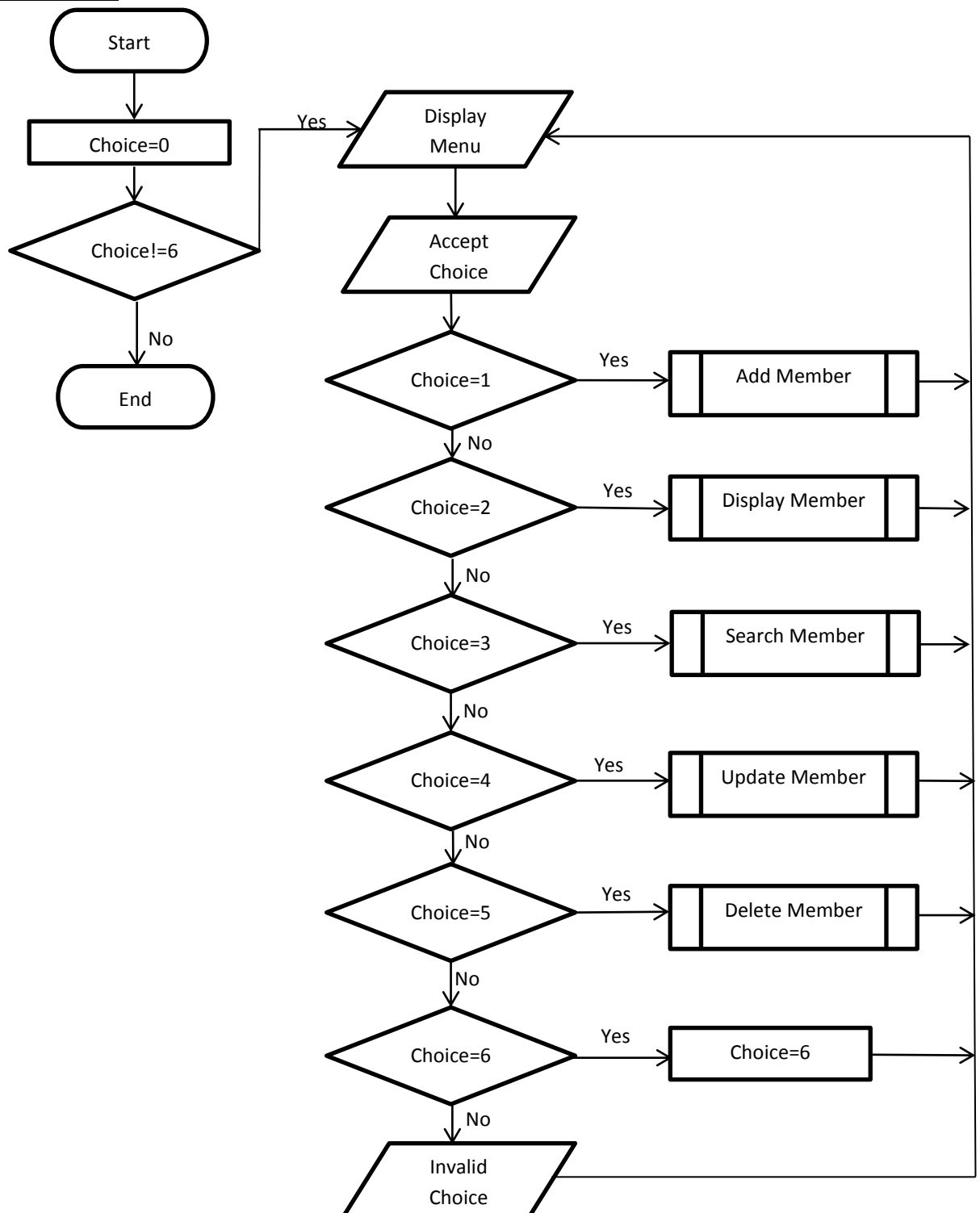
❖ Update Expiry Date

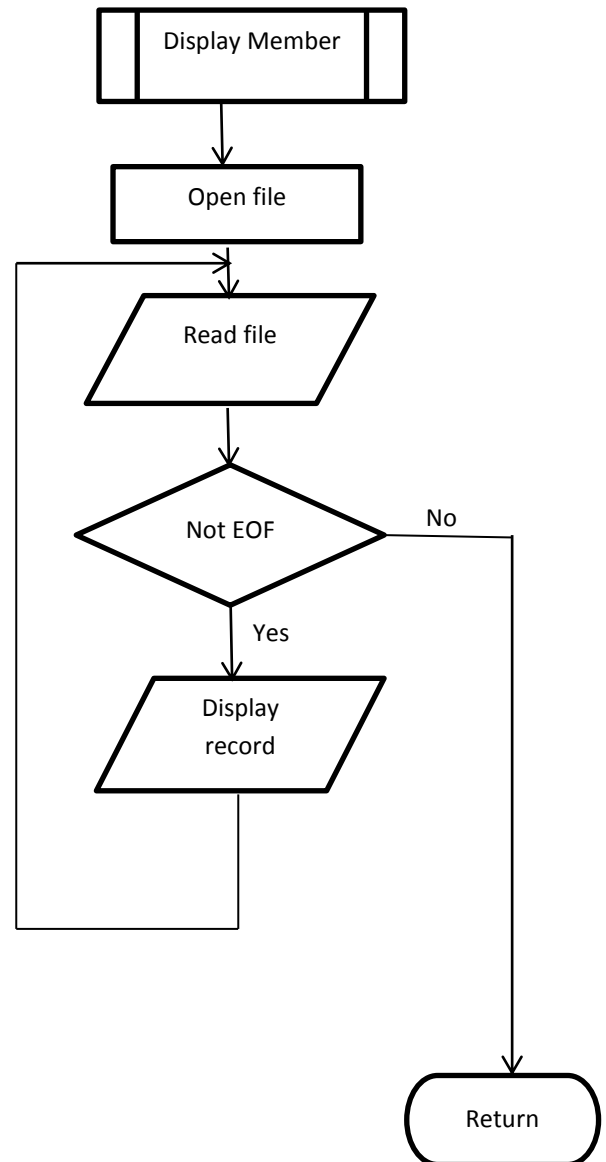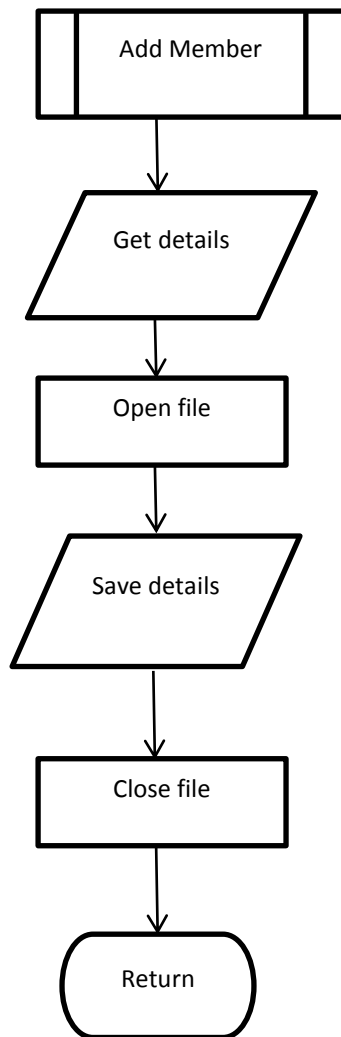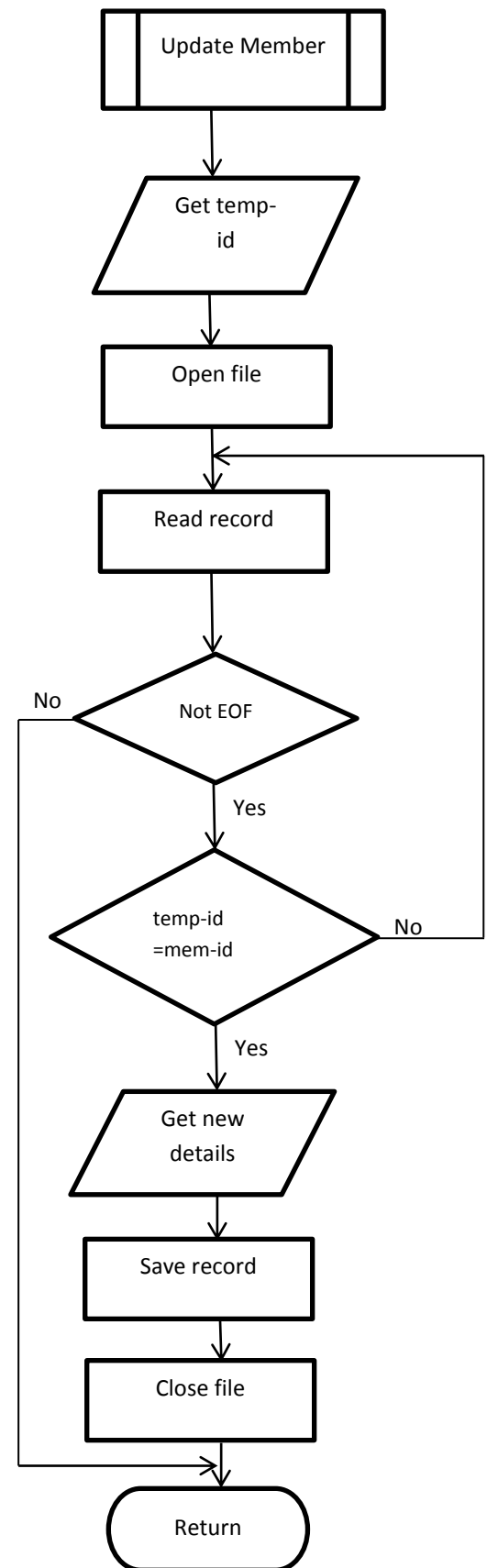**5. Delete Member Record**

❖ Delete record using Member ID

6. Exit

❖ Exit from the program

2. PROGRAM DESIGN

2.1 Flowcharts

Add Member

Get details

↓

Open file

↓

Save details

↓

Close file

↓

Return

Display Member

Open file

↓

Read file
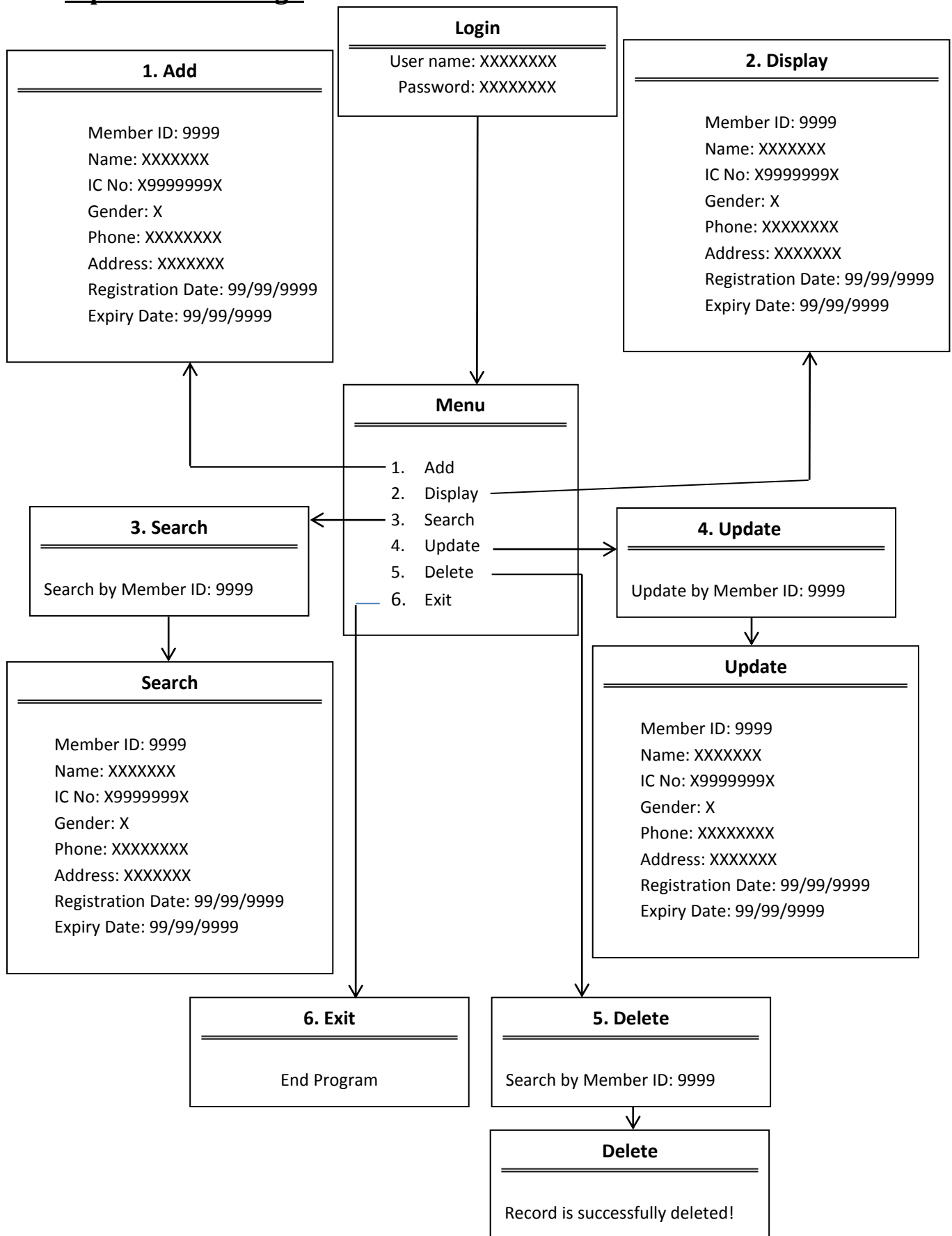
↓

Not EOF

No →

Yes ↓

Display record

Return

Search Member

Get temp-id

Open file

Error — Yes

No

Read record

Not EOF — No → Return

Yes

temp-id =mem-id — No

Yes

Display record

Read record

Update Member

Get temp-id

Open file

Read record

Not EOF — No

Yes

temp-id =mem-id — No

Yes

Get new details

Save record

Close file

Return

Delete Member

Open existing file

Error

Yes

Display Error Message

No

Get temp-id

Open temp file

Error

Yes

Display Error Message

No

Read existing file

Not EOF

No

Yes

temp-id =mem-id

No

check=1

Yes

Save to temp file

Close all files

Read existing file

Rename files

Return

2.2 Input Screen Design

Login
═══════════════════
User name: XXXXXXX
Password: XXXXXXXX

1. Add
═══════════════════

Member ID: 9999
Name: XXXXXXX
IC No: X9999999X
Gender: X
Phone: XXXXXXXX
Address: XXXXXXX
Registration Date: 99/99/9999
Expiry Date: 99/99/9999

2. Display
═══════════════════

Member ID: 9999
Name: XXXXXXX
IC No: X9999999X
Gender: X
Phone: XXXXXXXX
Address: XXXXXXX
Registration Date: 99/99/9999
Expiry Date: 99/99/9999

Menu
═══════════════════

1. Add
2. Display
3. Search
4. Update
5. Delete
6. Exit

3. Search
═══════════════════

Search by Member ID: 9999

4. Update
═══════════════════

Update by Member ID: 9999

Search
═══════════════════

Member ID: 9999
Name: XXXXXXX
IC No: X9999999X
Gender: X
Phone: XXXXXXXX
Address: XXXXXXX
Registration Date: 99/99/9999
Expiry Date: 99/99/9999

Update
═══════════════════

Member ID: 9999
Name: XXXXXXX
IC No: X9999999X
Gender: X
Phone: XXXXXXXX
Address: XXXXXXX
Registration Date: 99/99/9999
Expiry Date: 99/99/9999

6. Exit
═══════════════════

End Program

5. Delete
═══════════════════

Search by Member ID: 9999

Delete
═══════════════════

Record is successfully deleted!

3. PROGRAM SPECIFICATION

3.1 File Specification

No.	Item Name	Description	Data Type	Size	Range
1	mem_id	Member ID	char	4	0-9999
2	name	Name	char	20	a-z/A-Z
3	ic_no	IC_No.	char	10	A-Z/9999999
4	gender	Gender	char	1	M/F
5	phone	Phone	char	10	0-9999999
6	addr	Address	char	30	a-z/A-Z/0-999999999
7	reg_date	Registration Date	char	10	12/12/9999
8	exp_date	Expiry Date	char	10	12/12/9999

3.2 Processing and Validation

1. mem_id

User must input 4 digits format. '0001'. Must not be a duplication. If so, error message will be shown.

2. name

Name can be entered any character.

3. ic_no

User must enter 9 characters-format (S1234567X). If not 9 characters error message will be displayed.

4. gender

Gender should be (M/m) or (F/f). Small or capital letter is acceptable.

5. phone

Phone must have 8 characters only. If not, error message will be displayed.

6. addr

Addr can be any character.

7. reg_date

Registration date must follow the format (DD/MM/YYYY).

8. exp_ date

Expiry date must follow the format (DD/MM/YYYY).

3.3 <u>Messages used in the program</u>

password function

If the user enters the correct password, "Login successful !" will be displayed. If the user enters the wrong password, "Login failed" will be shown. If the user reaches 3 times, the maximum tries limit, "Maximum try limits reached. System will be shut down" will be displayed.

main function

In main menu, user needs to put menu choice from 1 to 6. If other number or character is entered, "Invalid choice ! Press any key to continue …" will be shown up.

add_mem function

In add member, if there is no file existed, "Cannot open output file" will be shown. If the member enters a duplicate ID, "Duplicate ID. Please enter a unique ID" will be popped up. When entering IC No., if the input is not 9 characters, "IC No. must have 9 characters only" will be shown. When entering phone no., the input must be 8 characters, if not, "Phone must have 8 characters only" will be displayed. If the record has been added, "The record has been successfully added." will be shown.

display_mem function

If there is no file existed, "Cannot open output file" will be shown. If there is no record in the file, "No record in member database" will be shown.

search_mem function

If there is no file existed, "Cannot open output file" will be shown. If there is no record in the file, "No record in member database" will be shown.

update_mem function

If there is no file existed, "Cannot open output file" will be shown. If there is no record in the file, "No record in member database" will be shown. When entering IC No., if the input is not 9 characters, "IC No. must have 9 characters only" will be shown. When entering phone no., the input must be 8 characters, if not, "Phone must have 8 characters only" will be displayed. If the record has been updated, "The record has been successfully updated." will be shown.

delete_mem function

If there is no file existed, "Cannot open output file" will be shown and if there is no temporary file, "Cannot open temp file" will be displayed. If invalid ID is entered, "Unable to find the member ID" will be shown. When the user is about to delete the record, the confirmation message "Are you sure you want to delete[Y/N] ?" will be asked and after deleting of record, "Record is successfully deleted" will be popped up.

4. DEVELOPMENT AND TESTING

4.1 Program Code

```c
/* Create Library Membership Control System */


#include<stdio.h>        //declare prototypes

#include<conio.h>

#include<string.h>

#include<ctype.h>

#include<stdlib.h>


void add_mem(void);   //declare functions

void display_mem(void);

void update_mem(void);

void delete_mem(void);

void search_mem(void);

void password(void);


struct                   //declare variables in struct
{
        char mem_id[5];

        char name[20];

        char ic_no[10];

        char gender;

        char phone[10];

        char addr[30];

        char reg_date[11];

        char exp_date[11];
}member;
```

```c
FILE *fptr;              //declare file


void main(void)                 //main function
{
        int choice=0;
        password();
        while(choice!=6)
        {
            clrscr();
            gotoxy(24,3);printf("Library Membership Control System");
            gotoxy(24,4);printf("*****************************\n\n\n");
            gotoxy(30,7);printf("1. Add Memeber\n");//construct menu
            gotoxy(30,8);printf("2. Display All Member\n");
            gotoxy(30,9);printf("3. Search Member\n");
            gotoxy(30,10);printf("4. Update Member\n");
            gotoxy(30,11);printf("5. Delete Member\n");
            gotoxy(30,12);printf("6. Exit\n\n\n");
            gotoxy(26,14);printf("Please enter choice !!! ");
            gotoxy(50,14);scanf("%d",&choice);

            switch(choice)
            {
                    case 1: add_mem();
                    break;
                    case 2: display_mem();
                    break;
                    case 3: search_mem();
                    break;
                    case 4: update_mem();
                    break;
```

```
                case 5: delete_mem();

                break;

                case 6: choice=6;

                break;

                default:

                gotoxy(21,16);printf("Invalid choice ! Press any key to continue...");

                getch();

        }                               //end of switch case

    }                       //end of while loop
}                           //end of main program


void add_mem(void)                  //add_member function
{
        char temp_id[5];


        clrscr();
        printf("Enter Member ID ");
        fflush(stdin);
        gets(temp_id);


        if((fptr=fopen("D:library.dat","ab+"))==NULL)               //open file
        {
                fprintf(stderr,"Cannot open output file.\n");
                getch();
        }


        fseek(fptr,SEEK_SET,0);
        fread(&member,sizeof(member),1,fptr);


        while(!feof(fptr))
```

```c
        {
                while(strcmp(member.mem_id,temp_id)==0)
                {
                        printf("Duplicate ID !!!");
                        printf("\nPlease enter a unique ID");
                        getch();


                        clrscr();
                        printf("Enter Member ID ");
                        fflush(stdin);
                        gets(temp_id);
                }
                fread(&member,sizeof(member),1,fptr);
        }

        strcpy(member.mem_id,temp_id);
        fclose(fptr);


        printf("Enter Name ");
        fflush(stdin);
        gets(member.name);



 do{
                printf("Enter IC no: ");
                fflush(stdin);
                gets(member.ic_no);

                if(strlen(member.ic_no)!=9)
                {
```

```
                    printf("IC No. must have 9 characters only.\n");

            }


        }while(strlen(member.ic_no)!=9);



        printf("Enter gender(M/F) ");
        scanf("%c",&member.gender);
        member.gender=toupper(member.gender);


        do{
                printf("Enter phone no: ");
                fflush(stdin);
                gets(member.phone);


                if(strlen(member.phone)!=8)
                        printf("Phone must have 8 characters only.\n");

  }while(strlen(member.phone)!=8);


        printf("Enter address ");
        fflush(stdin);
        gets(member.addr);


        printf("Enter registration date (DD/MM/YYYY) ");
        fflush(stdin);
        gets(member.reg_date);


        printf("Enter expiry date (DD/MM/YYYY) ");
        fflush(stdin);
```

```
        gets(member.exp_date);


        if((fptr=fopen("D:library.dat","ab+"))==NULL)          //open file
        {
               fprintf(stderr,"Cannot open output file.\n");
               getch();
        }


        fwrite(&member,sizeof(member),1,fptr);                 //write to file
        fclose(fptr);                              //close file


        printf("Record has been successfully added.");
        getch();
}

void display_mem(void)                            //display member function
{
        clrscr();
        if((fptr=fopen("D:library.dat","rb+"))==NULL)     //open file
        {
               fprintf(stderr,"Cannot open output file.\n");
               getch();
        //       exit(0);
        }


        fseek(fptr,SEEK_SET,0);                         //set pointer to position 0
        fread(&member,sizeof(member),1,fptr);           //read record


        if(feof(fptr))
        {
```

```c
        gotoxy(30,1);

        printf("No record in Member Database");

        getch();

    }


    while(!feof(fptr))                          //while not end of file
    {

        printf("Member ID:  %s\n",member.mem_id);

        printf("Name: %s\n",member.name);

        printf("IC No: %s\n",member.ic_no);

        printf("Gender: %c\n",member.gender);

        printf("Phone: %s\n",member.phone);

        printf("Address: %s\n",member.addr);

        printf("Registration Date: %s\n",member.reg_date);

        printf("Expiry Date: %s\n \n",member.exp_date);


        fread(&member,sizeof(member),1,fptr);        //read record

    }
    fclose(fptr);                               //close file

    getch();
}


void search_mem(void)                                //search record
{

    char temp_id[5];


    clrscr();

    printf("Enter Member ID to be searched ");   //prompt to accept ID to search

    fflush(stdin);

    gets(temp_id);
```

```c
if((fptr=fopen("D:library.dat","rb+"))==NULL)         //open file
{
        fprintf(stderr,"Cannot open output file\n");
        getch();
}


fseek(fptr,SEEK_SET,0);                                //set pointer to position 0
fread(&member,sizeof(member),1,fptr);          //read record


if(feof(fptr))
{
        gotoxy(30,1);
        printf("No record in Member Database");
        getch();
}


while(!feof(fptr))                              //while not end of file
{
        if(strcmp(temp_id,member.mem_id)==0)      //compare 2 IDs and show record if found
        {
                printf("Name: %s \n",member.name);
                printf("IC No: %s \n",member.ic_no);
                printf("Gender: %c \n",member.gender);
                printf("Phone: %s \n",member.phone);
                printf("Address: %s \n",member.addr);
                printf("Registration Date: %s \n",member.reg_date);
                printf("Expiry Date: %s \n",member.exp_date);
        }
        fread(&member,sizeof(member),1,fptr);            //read record
```

```c
        }
        fclose(fptr);                                      //close file
        getch();
}


void update_mem(void)                                      //update record
{
        char temp_id[5];
        int filepos;


        clrscr();
        printf("Enter Member ID to be updated ");    //prompt user to accept ID to search
        fflush(stdin);
        gets(temp_id);


        if((fptr=fopen("D:library.dat","rb+"))==NULL)              //open file
        {
                fprintf(stderr,"Cannot open output file\n");
                getch();
        }


        fseek(fptr,SEEK_SET,0);                             //set pointer to position 0
        fread(&member,sizeof(member),1,fptr);               //read record


        if(feof(fptr))
        {
                gotoxy(30,1);
                printf("No record in Member Database");
                getch();
        }
```

```c
while(!feof(fptr))                              //while not end of file
{
        if(strcmp(temp_id,member.mem_id)==0)  //compare 2 IDs and if it's found
        {

                printf("Enter new Name ");           //accept new values
                fflush(stdin);
                gets(member.name);


                do{
                        if(strlen(member.ic_no)!=9)
                        {
                                printf("Enter new IC no: ");
                                fflush(stdin);
                                gets(member.ic_no);


                                printf("\nIC No. must be 9 characters only\n");
                        }

                }while(strlen(member.ic_no)!=9);

                printf("Enter new Gender(M/F) ");
                scanf("%c",&member.gender);
                member.gender=toupper(member.gender);

                do{
                        if(strlen(member.phone)!=8)
                        {
                                printf("Enter new phone no: ");
```

```c
                                fflush(stdin);

                                gets(member.phone);


                                printf("\nPhone must be 8 characters only\n");
                        }
                }while(strlen(member.phone)!=8);


                printf("Enter new address ");
                fflush(stdin);
                gets(member.addr);


                printf("Enter new registration date (DD/MM/YYYY) ");
                fflush(stdin);
                gets(member.reg_date);


                printf("Enter new expiry date (DD/MM/YYYY) ");
                fflush(stdin);
                gets(member.exp_date);


                filepos=ftell(fptr)-sizeof(member);
                fseek(fptr,filepos,0);


                fwrite(&member,sizeof(member),1,fptr);
                fclose(fptr);


                printf("Record is updated successfully.");
        }
        fread(&member,sizeof(member),1,fptr);                    //read record
    }
    fclose(fptr);                                               //close file
```

```
        getch();
}


void delete_mem(void)                                //delete record
{
        FILE *temp;
        char confirm;
        char tempname[80]="D:temp.dat",existingname[80]="D:library.dat";
        int check;
        char del_id[5];


        clrscr();


        if((fptr=fopen("D:library.dat","rb+"))==NULL)  //open existing file in read mode
        {
                printf("Cannot open input file.");
                getch();
        }


        check=0;
        clrscr();
        printf("Please enter Member ID to be deleted! ");          //prompt user the ID to be
        deleted
        fflush(stdin);
        gets(del_id);


        if((temp=fopen("D:temp.dat","wb+"))==NULL)                 //open temp file in write
        mode
        {
                printf("Cannot open temporary file.");
                getch();
```

```
        }


        fseek(fptr,SEEK_SET,0);                              //set pointer posistion to
0

        fread(&member,sizeof(member),1,fptr);                //read from existing file


        while(!feof(fptr))                                   //while not end of file
        {
                if(strcmp(del_id,member.mem_id)==0)          //compare 2 IDs and if it's
        found
                        check=1;
                else                                         //if the ID is not found
                {
                        fwrite(&member,sizeof(member),1,temp);        //write to temp file
                }


                fread(&member,sizeof(member),1,fptr);        //read from existing file
        }
        fclose(fptr);                                        //close existing file
        fclose(temp);                                        //close temp file


        if(check!=1)
        {
            clrscr();
            printf("Unable to find the Member ID %s",del_id);   //if ID is not found show Not
        found
            getch();
        }
        else
        {
            clrscr();                                        //else ask for confirmation
            printf("Are you sure you want to delete[Y/N] ?");
```

```c
            fflush(stdin);

            scanf("%c",&confirm);


            if(toupper(confirm)=='Y')                       //if confirm, remove existing file
      and
            {
                  remove(existingname);

                  rename(tempname,existingname);                 //rename temp file to
      existing file name

                  printf("Record is successfully deleted!");

                  getch();

            }
      }
      remove(tempname);                                     //remove temp file
}


void password(void)
{
      int i=0;
      char pass[8]="octopus";
      char tmp[8];


      clrscr();
      printf("\n \nWelcome to Library Membership Control System\n");
      printf("--------------------------------------------\n\n\n");
      printf("System Login\n \n");
      while(i<3)
      {
            printf("Enter password  ");
            fflush(stdin);
            gets(tmp);
```

```
        if(strcmp(tmp,pass)==0)

        {

                printf("Login successful !");

                getch();

                break;

        }

        else

        {

                printf("\nLogin failed !\n\n");

                getch();

        }

        i++;

    }

    if(i==3)

    {

        printf("\nMaximum try limits reached  !");

        printf("\n\nSystem will be shut down !!!");

        getch();

        exit(0);

    }

}
```

4.2 <u>Program Testing</u>

4.2.1 Test Plan
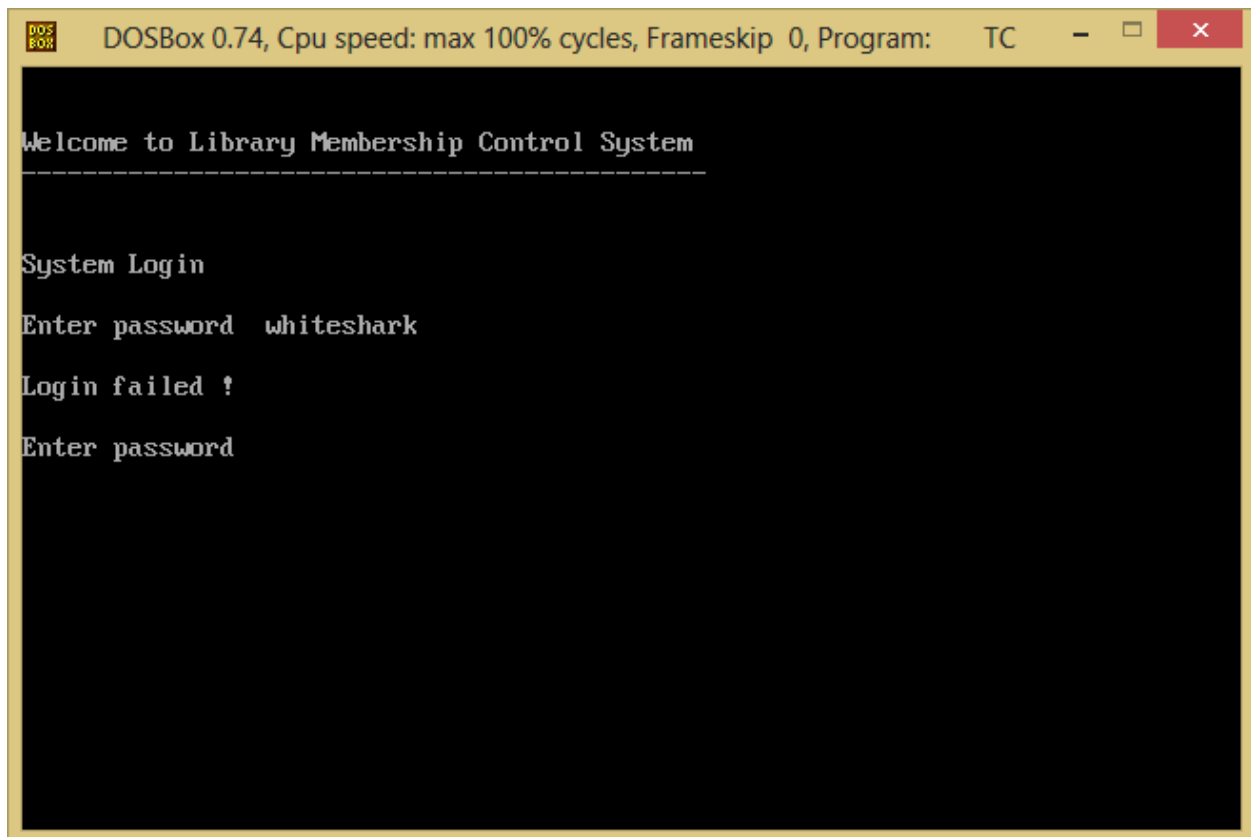
Test	Objectives
Case 1	To check whether it prompts up an error message 'Login failed' when a user enters wrong username or password.

Case 2	To check whether the system is totally shut down when user enters wrong username or password for 3 times.
Case 3	To check whether it shows "Login successful" and goes to main menu if the right user name and password are entered.
Case 4	To check whether it shows the Add New Member function if the user enters 1 in main menu choice.
Case 5	To check whether it prompt 'Enter name' if the user enters 4 characters in Member ID field.
Case 6	To check whether it prompt 'Enter IC No.' after the user enters name in Name field.
Case 7	To check whether it shows error message if the user enters more than 9 characters in IC No. field.
Case 8	To check whether it shows error message if the user enters less than 9 characters in IC No. field.
Case 9	To check whether it prompts 'Enter gender (M/F)' if the user enters 9 characters in IC No. field.
Case 10	To check whether it prompts 'Enter phone no:' if the user enters (M/m) or (F/f) character in Gender field.
Case 11	To check whether it shows error message if the user enters less than 8 characters in phone no. field.
Case 12	To check whether it shows error message if the user enters more than 8 characters in phone no. field.
Case 13	To check whether it prompts 'Enter address' if the user enters 8 characters in phone no: field.
Case 14	To check whether it prompts 'Enter registration date' if the user enters address field.
Case 15	To check whether it prompts 'Enter expiry date' if the user enters registration date field.
Case 16	To check whether it displays 'Record has been successfully added' if the user enters expiry date field.
Case 17	To check whether it shows error message if the user enters a duplicate ID in Member ID field.
Case 18	To check whether it shows the Display Member Details function if the

	user enters 2 in main menu choice.
Case 19	To check whether it shows error message if there is no record in the file when the user press 2 in main menu choice.
Case 20	To check whether it shows the Search Member function if the user enters 3 in main menu choice.
Case 21	To check whether it shows member details if the user enters available ID in search member function.
Case 22	To check whether it shows error message if there is no record in the file when the user enters member ID to be searched.
Case 23	To check whether it shows the Update Member function if the user enters 4 in main menu choice.
Case 24	To check whether it shows error message if there is no record in the file when the user enters member ID to be updated.
Case 25	To check whether it shows the Delete Member function if the user enters 5 in main menu choice.
Case 26	To check whether it shows error message if the user enters unavailable ID in delete member function.
Case 27	To check whether it shows confirm to be deleted message if the user enters available ID in delete member function.

4.2.2 Test Cases and Results

Test Case	1
Objectives	To check whether it prompts up an error message 'Login failed' when a user enters wrong username or password.
Test Data	'Enter User Name' & 'Enter Password'
Expected Test Result	If the wrong user name or password is entered, it will show 'Login failed' and prompt to enter for user name again.
Actual Test Result	As shown below
Conclusion	Successful. Error message was displayed when the wrong user name or password is entered and prompt to enter for user name again.
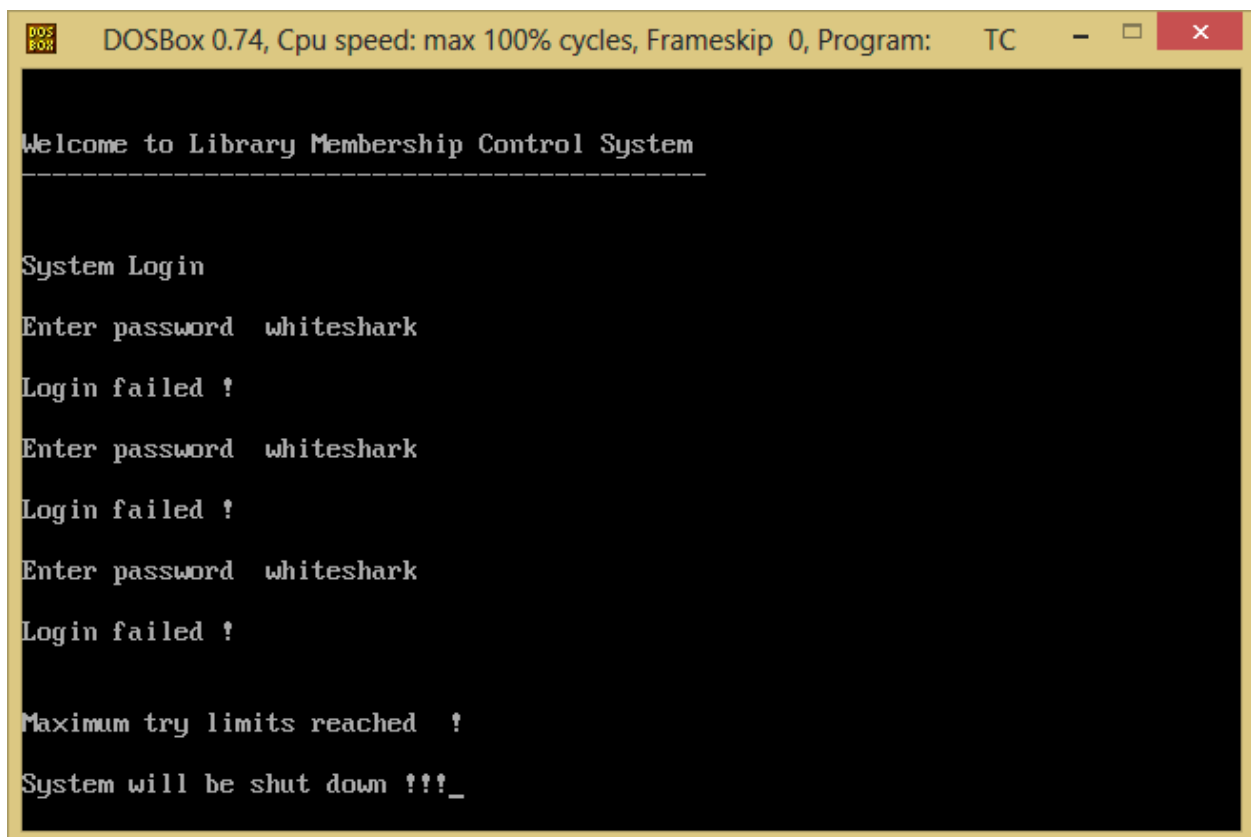
Test Case	2
Objectives	To check whether the system is totally shut down when user enters wrong username or password for 3 times.
Test Data	'Enter User Name' & 'Enter Password'
Expected Test Result	If the wrong user name or password is entered for 3 times, the system will be totally shut down.
Actual Test Result	As shown below
Conclusion	Successful. Error message was shown and the system was totally shut down when the wrong user name or password is entered for 3 times.

Test Case	3
Objectives	To check whether it shows "Login successful" and goes to main menu if the right user name and password are entered.
Test Data	'Enter User Name' & 'Enter Password'
Expected Test Result	If the right user name and password are entered, 'Login successful' and main menu will be displayed.
Actual Test Result	As shown below
Conclusion	Successful. 'Login successful' and main menu were shown when the right user name and password are entered.
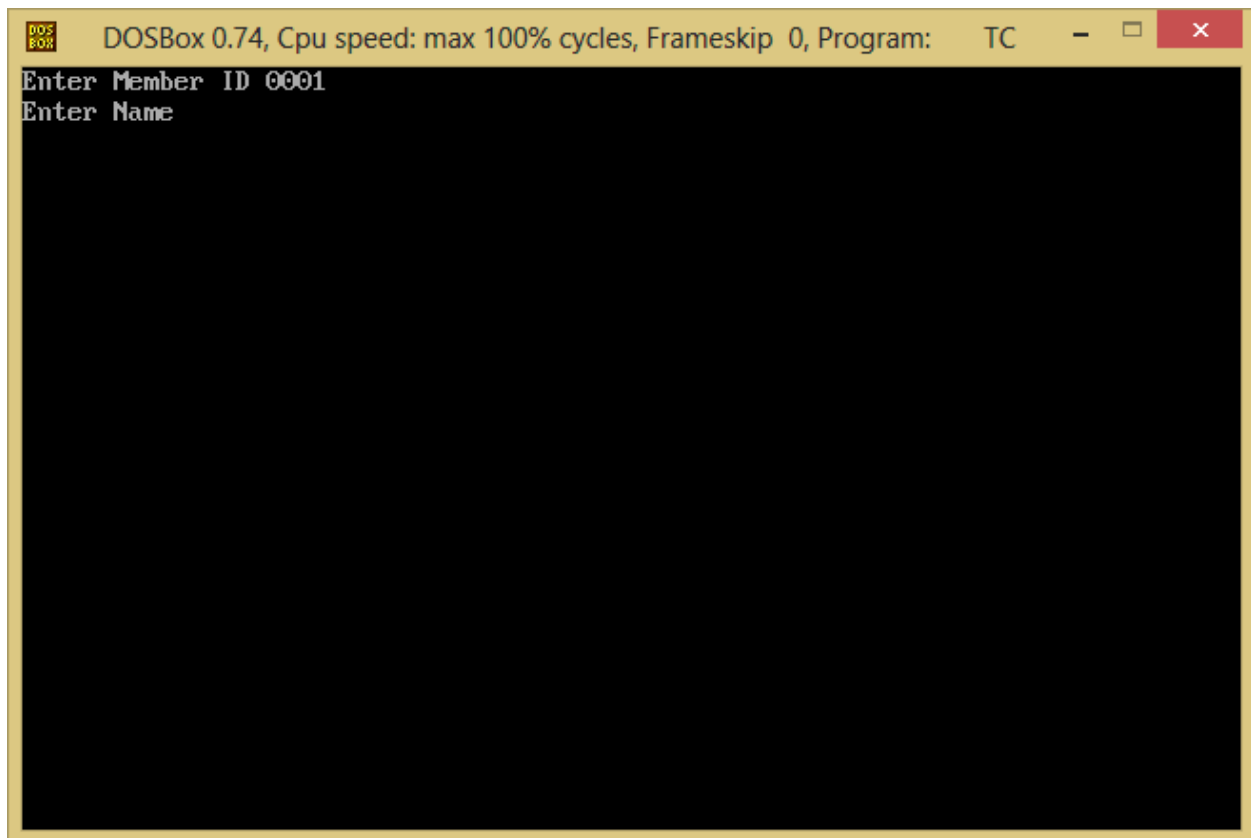
Test Case	4
Objectives	To check whether it shows the Add New Member function if the user enters 1 in main menu choice.
Test Data	Enter '1'
Expected Test Result	If the user enters 1, member ID will be prompted to enter.
Actual Test Result	As shown below
Conclusion	Successful. Enter Member ID was shown.
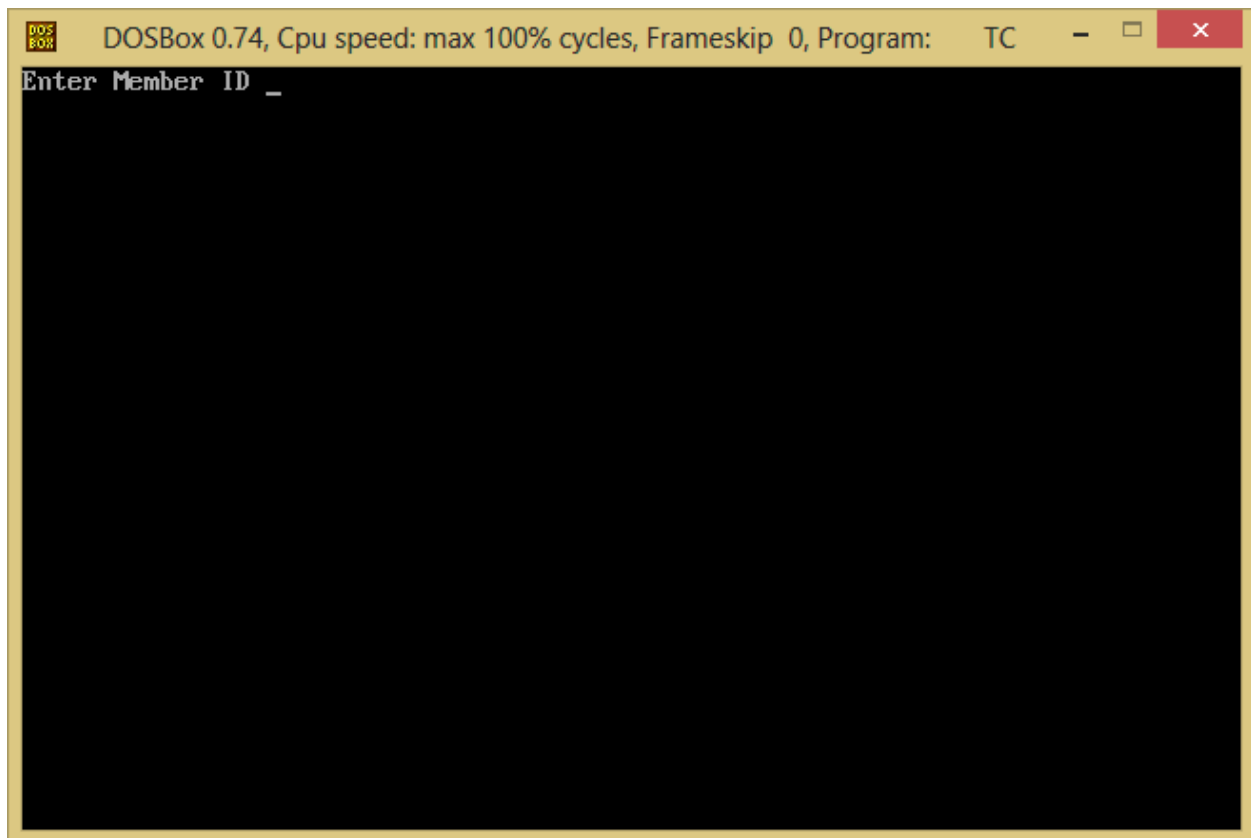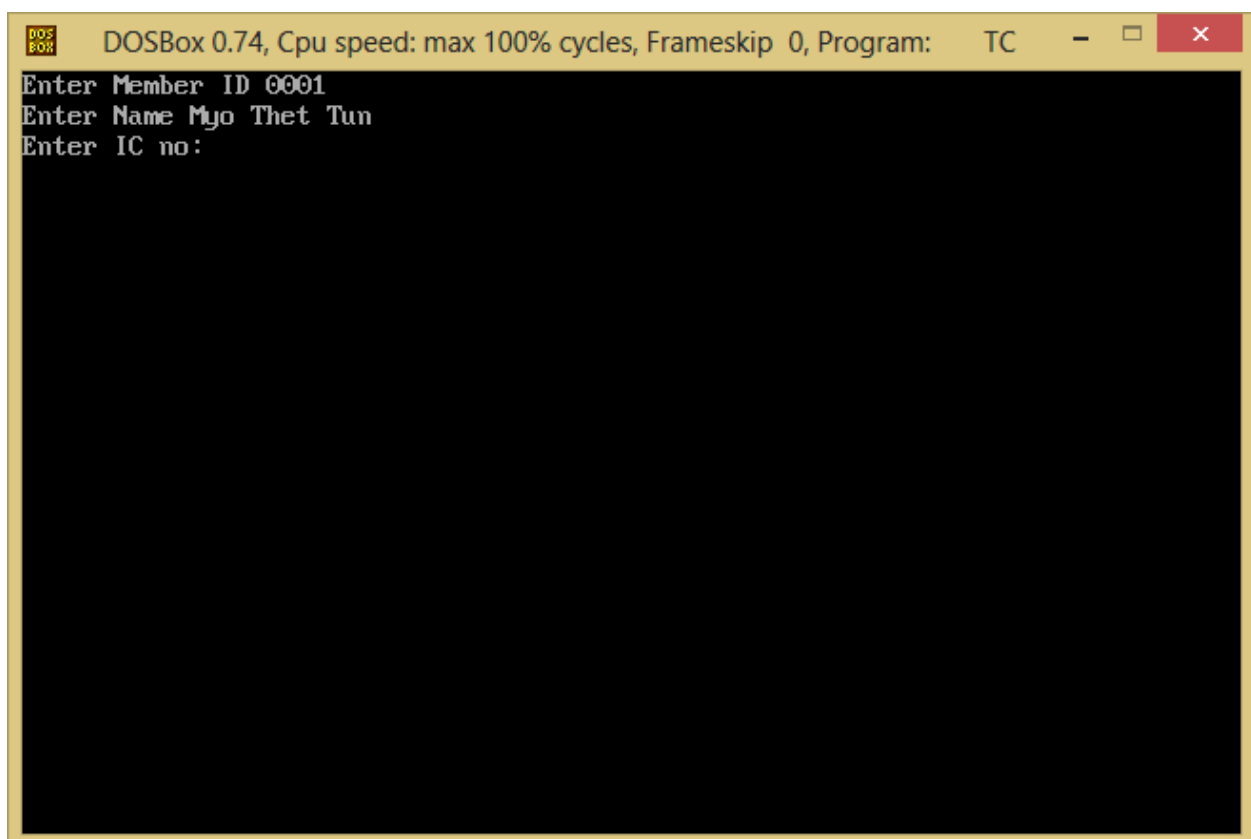
Test Case	5
Objectives	To check whether it prompt 'Enter name' if the user enters 4 characters in Member ID field.
Test Data	Enter '0001'
Expected Test Result	If the user enters 0001, 'Enter name' will be displayed.
Actual Test Result	As shown below
Conclusion	Successful. 'Enter name' was shown.
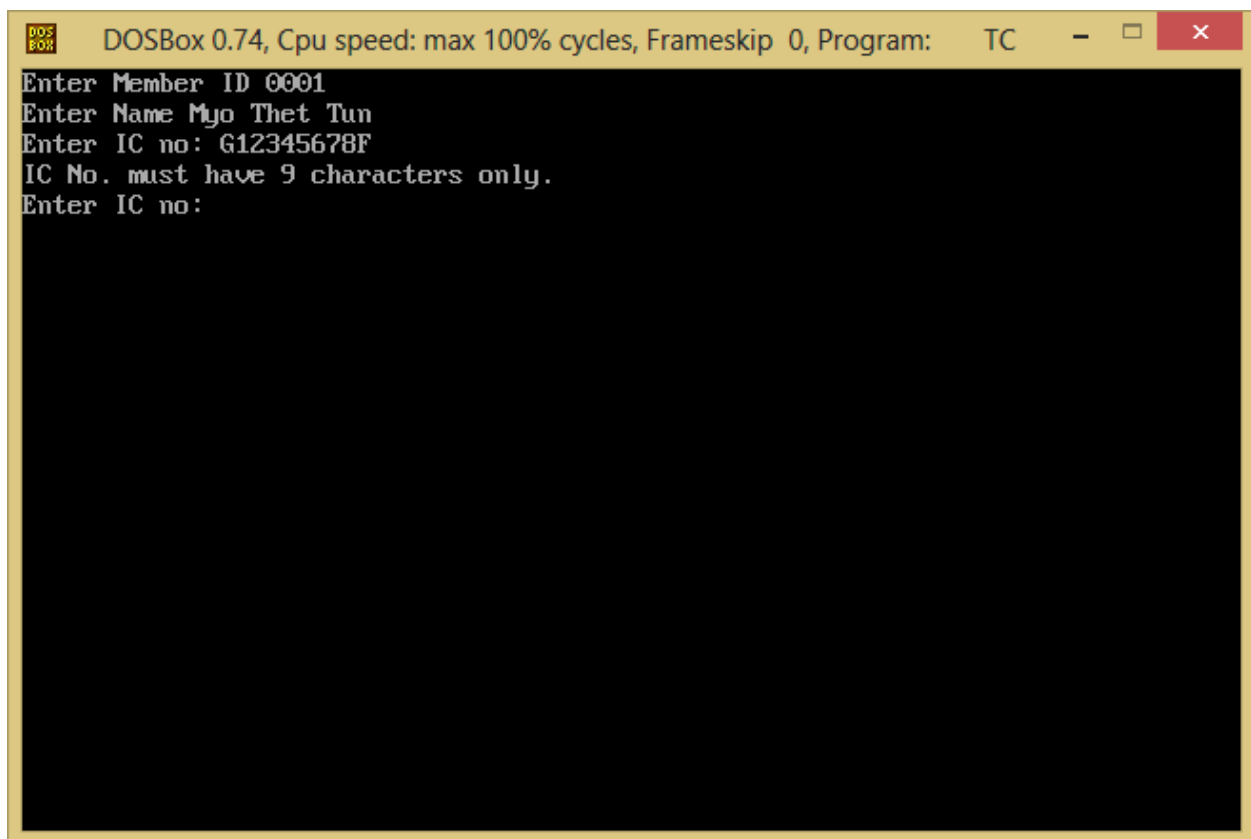
Test Case	6
Objectives	To check whether it prompt 'Enter IC No.' after the user enters name in Name field.
Test Data	Enter 'Myo Thet Tun'
Expected Test Result	If the user enters name, 'Enter IC No.' will be displayed.
Actual Test Result	As shown below
Conclusion	Successful. 'Enter IC No.' was shown.
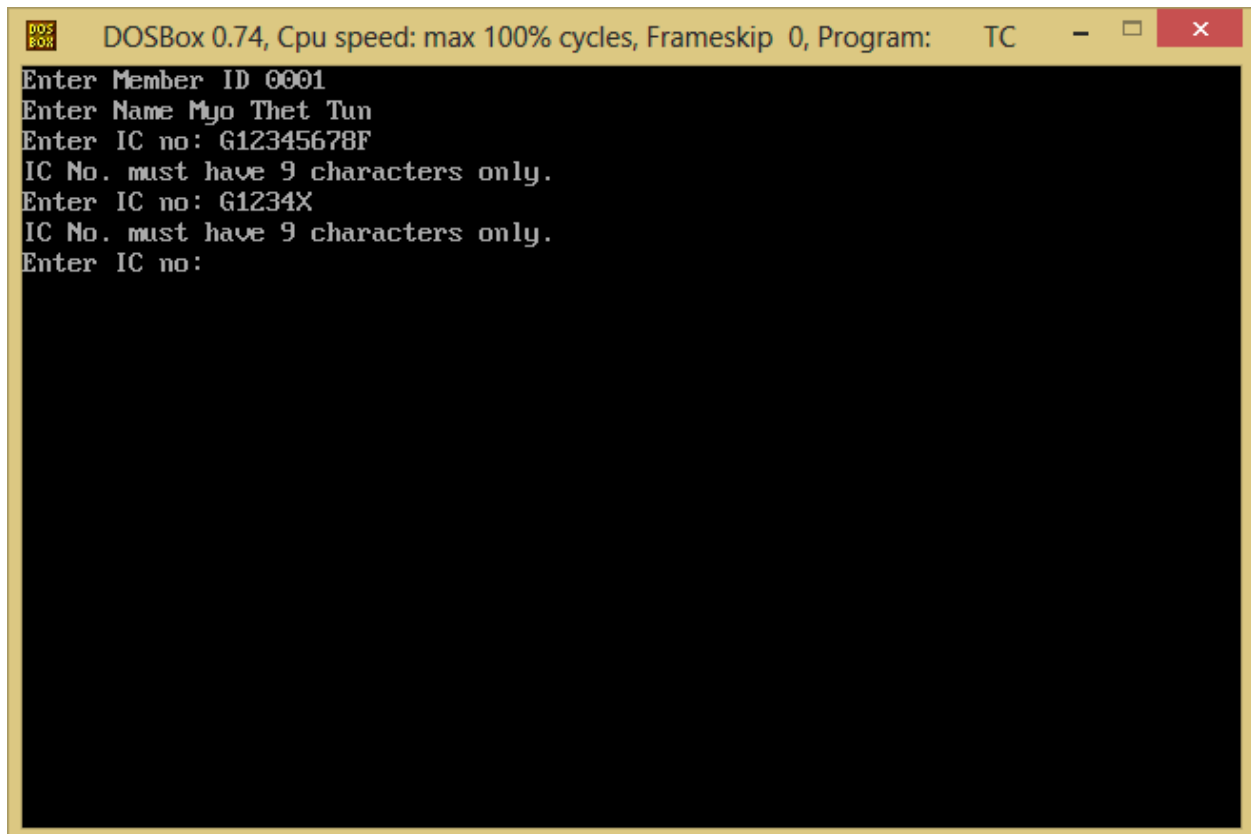
Test Case	7
Objectives	To check whether it shows error message if the user enters more than 9 characters in IC No. field.
Test Data	Enter 'G12345678F'
Expected Test Result	If the user enters 'G12345678F', error message will be displayed and IC No. will be prompted to enter again.
Actual Test Result	As shown below
Conclusion	Successful. Error message was shown and 'Enter IC No.' was shown again.
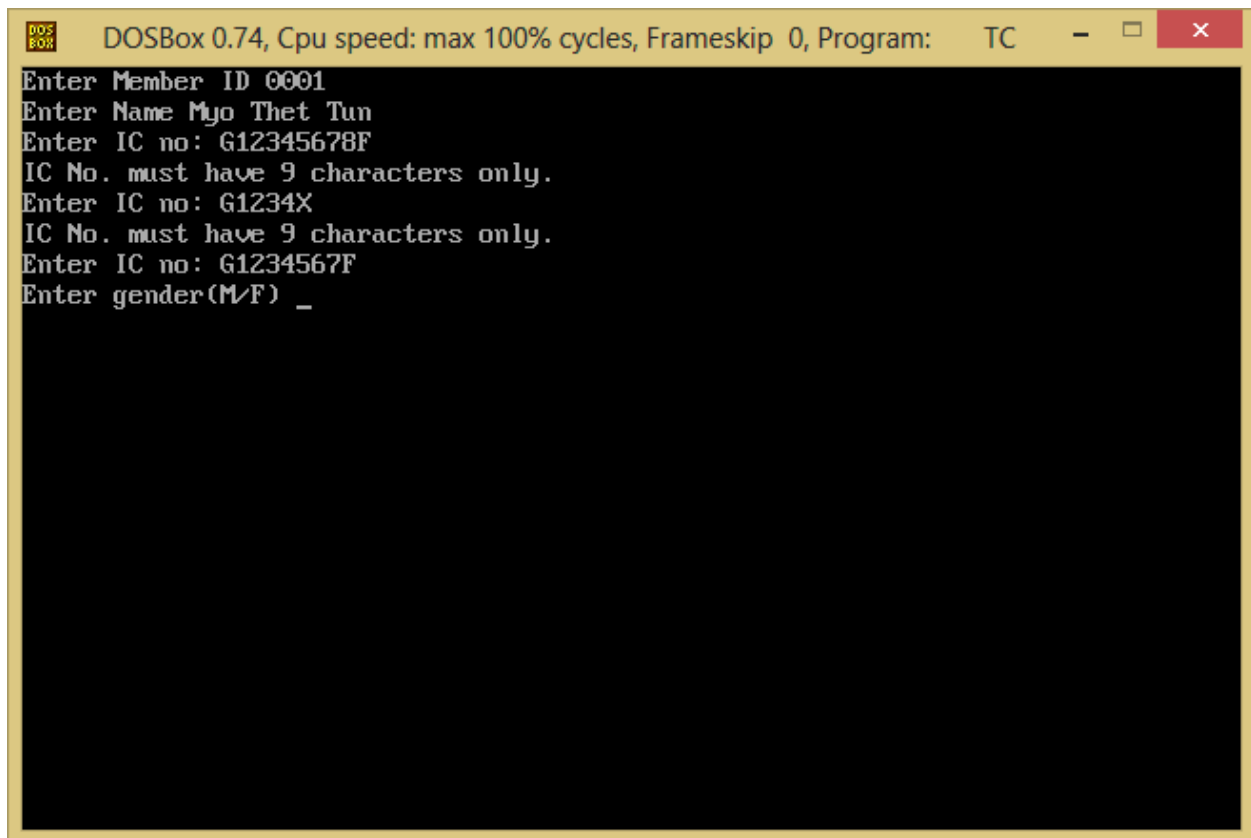
```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program:    TC
Enter Member ID 0001
Enter Name Myo Thet Tun
Enter IC no: G12345678F
IC No. must have 9 characters only.
Enter IC no:
```

Test Case	8
Objectives	To check whether it shows error message if the user enters less than 9 characters in IC No. field.
Test Data	Enter 'G1234X'
Expected Test Result	If the user enters 'G1234X', error message will be displayed and IC No. will be prompted to enter again.
Actual Test Result	As shown below
Conclusion	Successful. Error message was shown and 'Enter IC No.' was shown again.

Test Case	9
Objectives	To check whether it prompts 'Enter gender (M/F)' if the user enters 9 characters in IC No. field.
Test Data	Enter 'G1234567F'
Expected Test Result	If the user enters G1234567F, 'Enter gender (M/F)' will be displayed.
Actual Test Result	As shown below
Conclusion	Successful. 'Enter gender (M/F)' was shown.
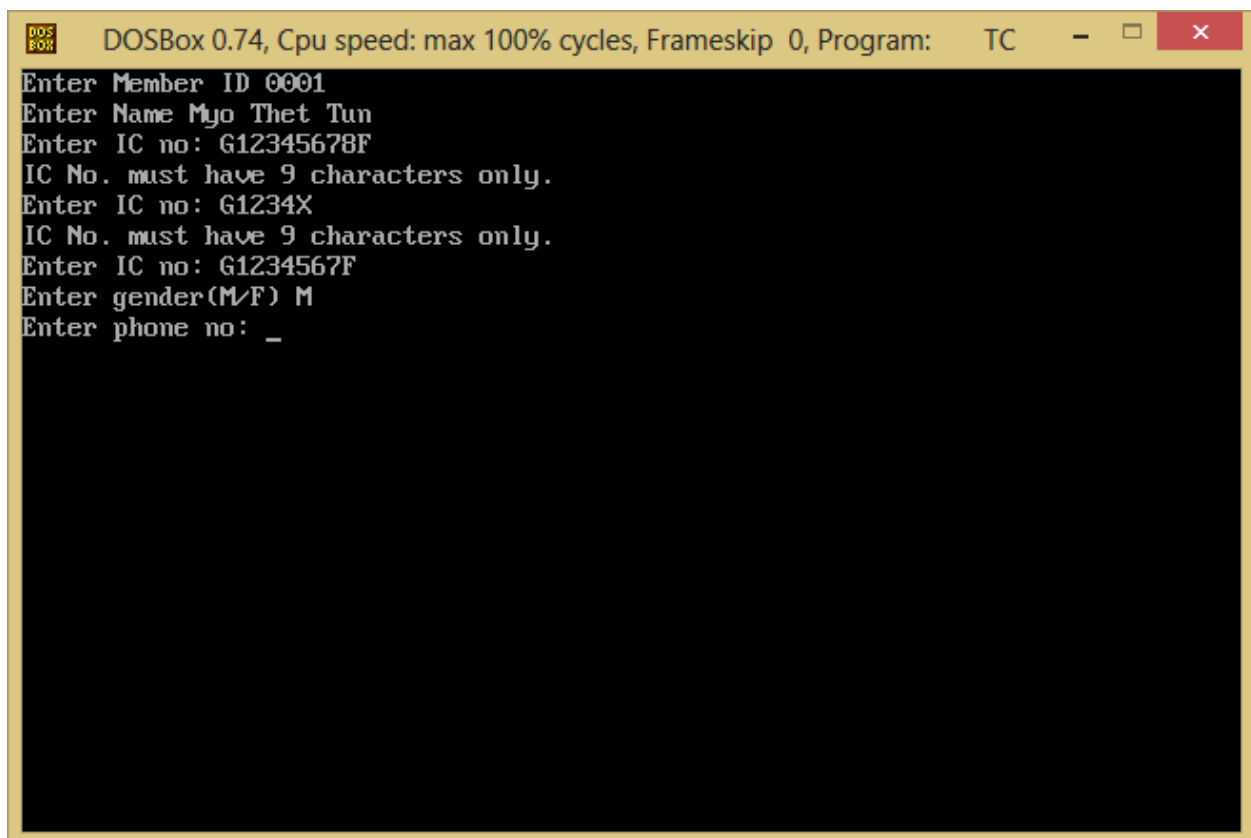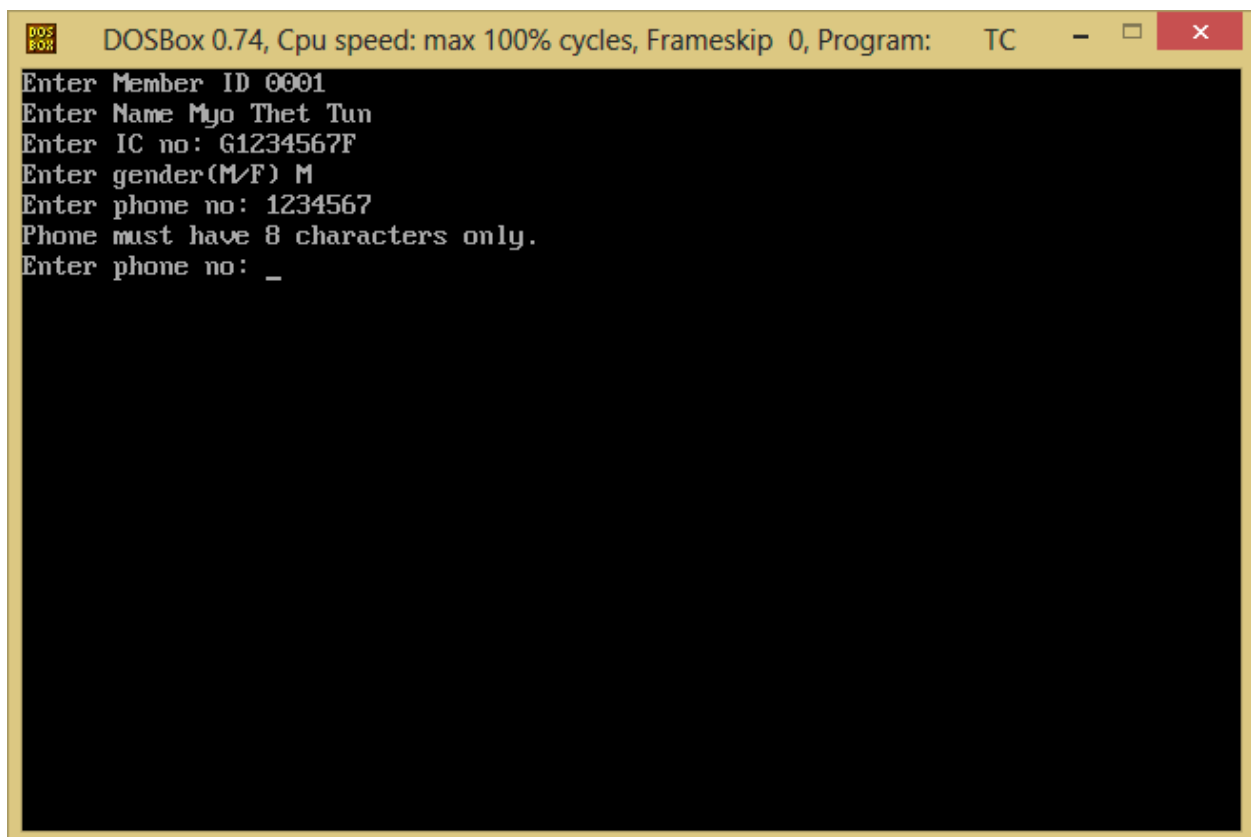
```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip  0, Program:    TC
Enter Member ID 0001
Enter Name Myo Thet Tun
Enter IC no: G12345678F
IC No. must have 9 characters only.
Enter IC no: G1234X
IC No. must have 9 characters only.
Enter IC no: G1234567F
Enter gender(M/F) _
```

Test Case	10
Objectives	To check whether it prompts 'Enter phone no:' if the user enters (M/m) or (F/f) character in Gender field.
Test Data	Enter 'M'
Expected Test Result	If the user enters M, 'Enter phone no:' will be displayed.
Actual Test Result	As shown below
Conclusion	Successful. 'Enter phone no:' was shown.

Test Case	11
Objectives	To check whether it shows error message if the user enters less than 8 characters in phone no. field.
Test Data	Enter '1234567'
Expected Test Result	If the user enters '1234567', error message will be displayed and phone no: will be prompted to enter again.
Actual Test Result	As shown below
Conclusion	Successful. Error message was shown and 'Enter phone no:' was shown again.

Test Case	12
Objectives	To check whether it shows error message if the user enters more than 8 characters in phone no. field.
Test Data	Enter '123456789'
Expected Test Result	If the user enters '123456789', error message will be displayed and phone no: will be prompted to enter again.
Actual Test Result	As shown below
Conclusion	Successful. Error message was shown and 'Enter phone no:' was shown again.
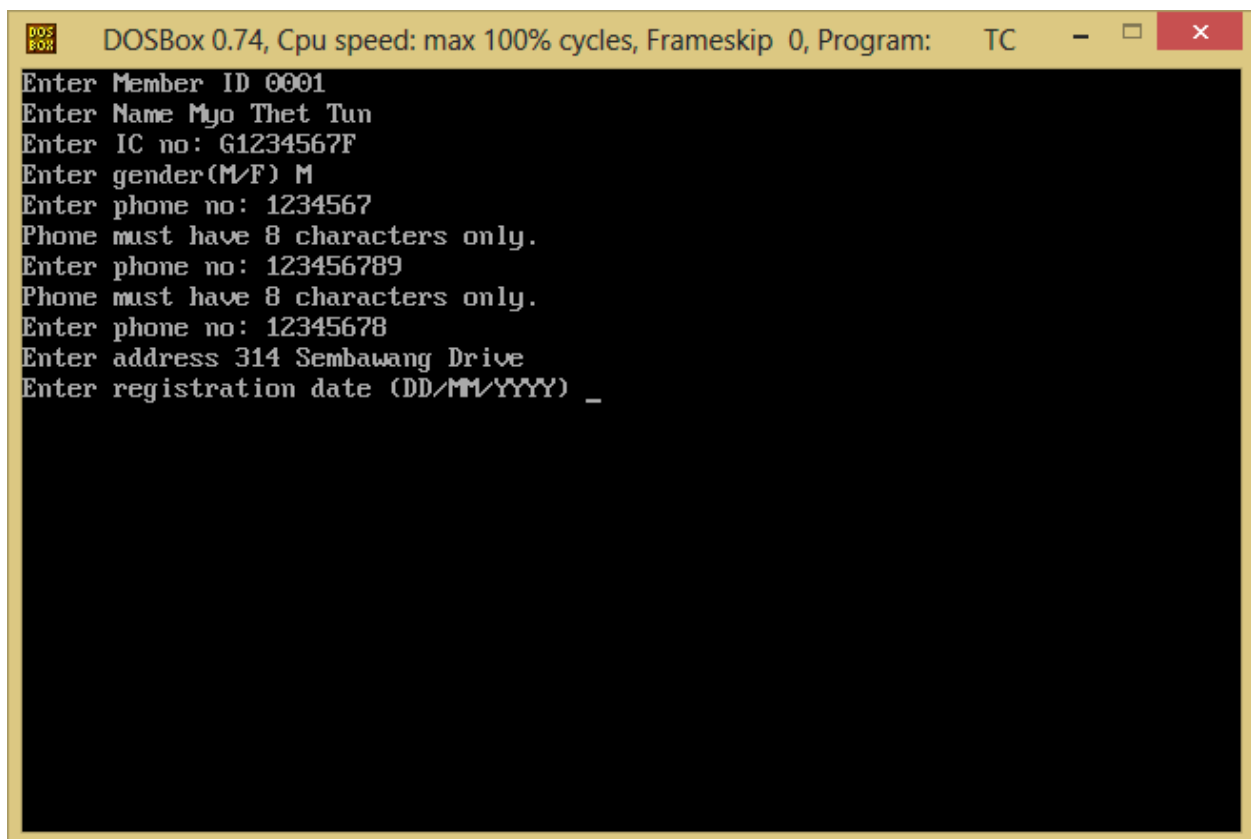
Test Case	13
Objectives	To check whether it prompts 'Enter address' if the user enters 8 characters in phone no: field.
Test Data	Enter '12345678'
Expected Test Result	If the user enters 12345678, 'Enter address' will be displayed.
Actual Test Result	As shown below
Conclusion	Successful. 'Enter address' was shown.

Test Case	14
Objectives	To check whether it prompts 'Enter registration date' if the user enters address field.
aTest Data	Enter '314 Sembawang Drive'
Expected Test Result	If the user enters address, 'Enter registration date' will be displayed.
Actual Test Result	As shown below
Conclusion	Successful. 'Enter registration date(dd/mm/yyyy)' was shown.
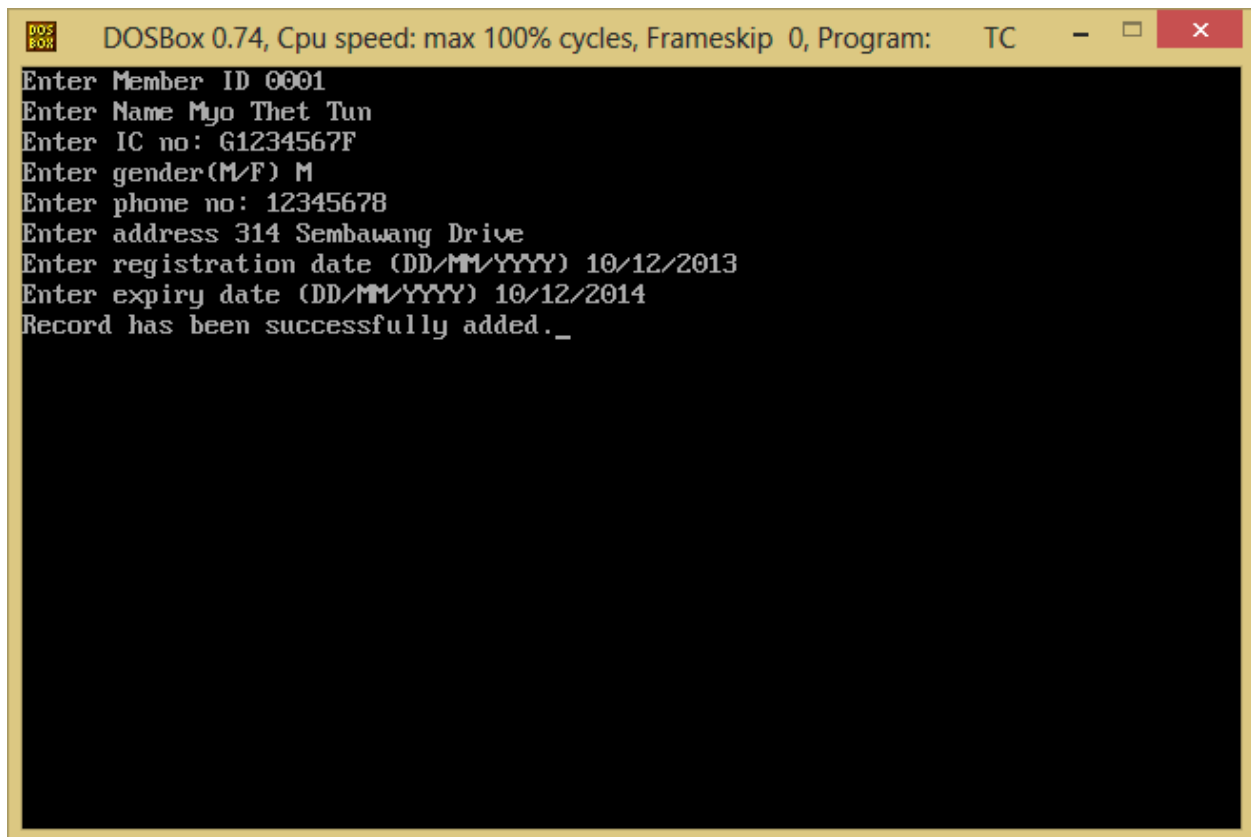
Test Case	15
Objectives	To check whether it prompts 'Enter expiry date' if the user enters registration date field.
aTest Data	Enter '10/12/2013'
Expected Test Result	If the user enters registration date, 'Enter expiry date' will be displayed.
Actual Test Result	As shown below
Conclusion	Successful. 'Enter expiry date (dd/mm/yyyy)' was shown.
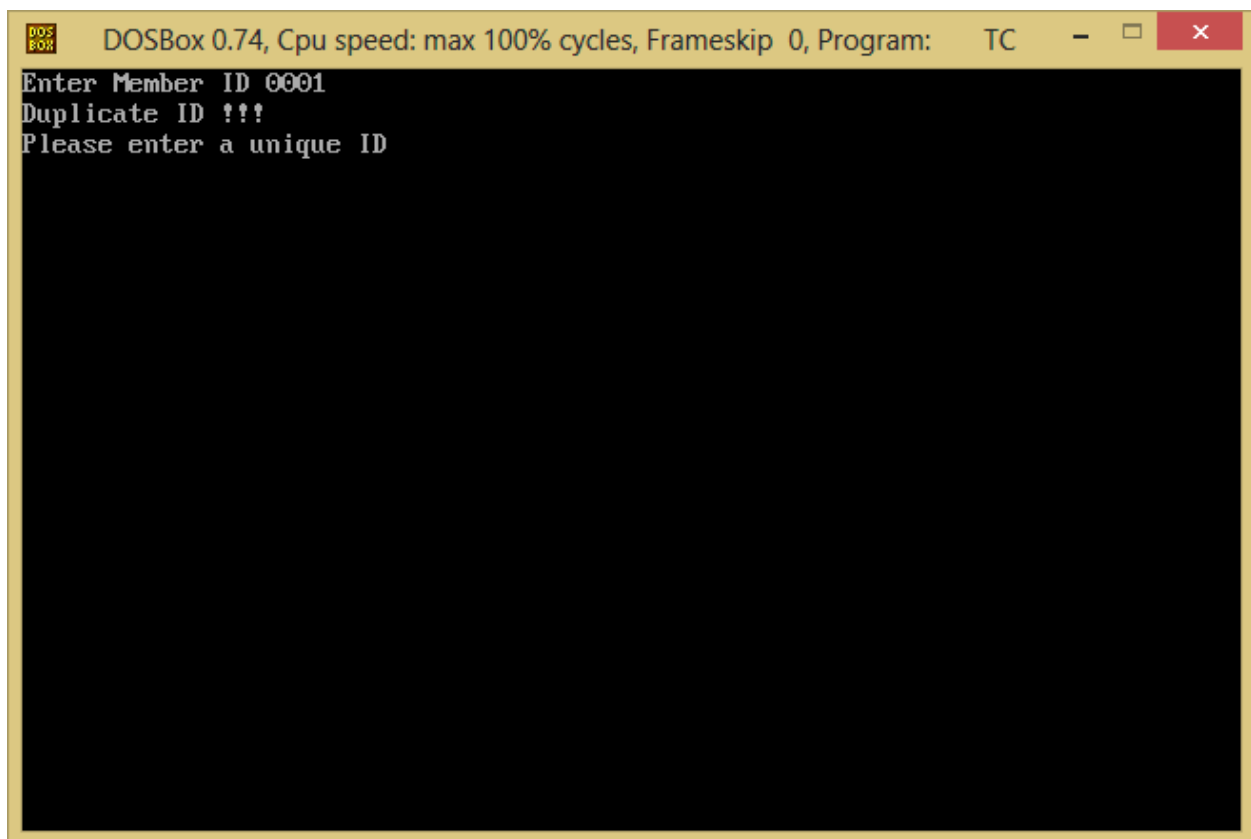
Test Case	16
Objectives	To check whether it displays 'Record has been successfully added' if the user enters expiry date field.
aTest Data	Enter '10/12/2014'
Expected Test Result	If the user enters expiry date, 'Record has been successfully added' will be displayed.
Actual Test Result	As shown below
Conclusion	Successful. 'Record has been successfully added.' was shown.
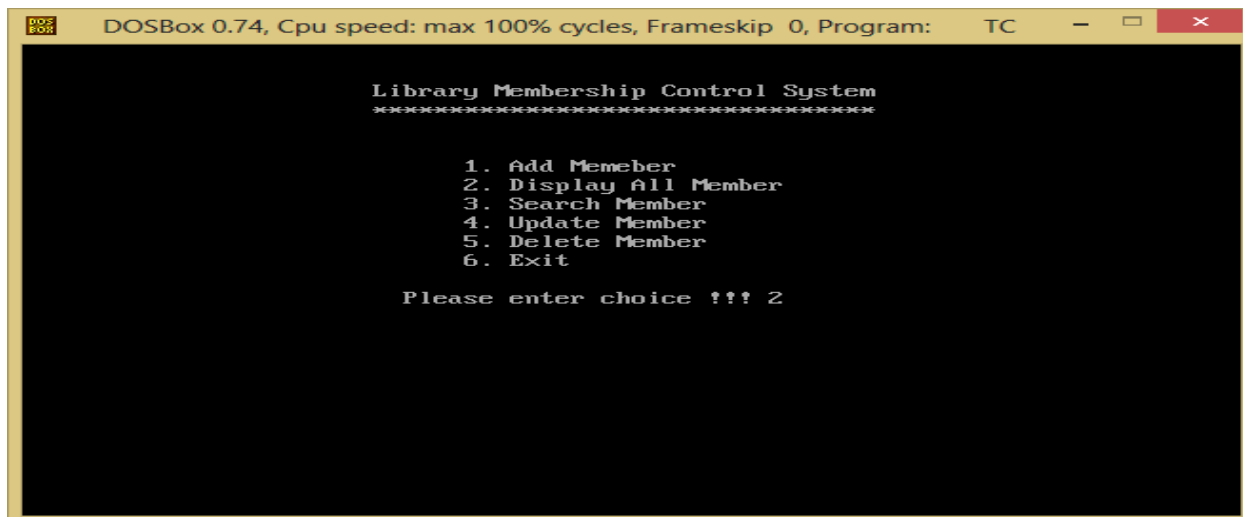
Test Case	17
Objectives	To check whether it shows error message if the user enters a duplicate ID in Member ID field.
Test Data	Enter '0001'
Expected Test Result	If the user enters '0001', 'Duplicated ID' error message will be displayed and Member ID will be prompted to enter again.
Actual Test Result	As shown below
Conclusion	Successful. Error message was shown and 'Enter Member ID.' was shown again.
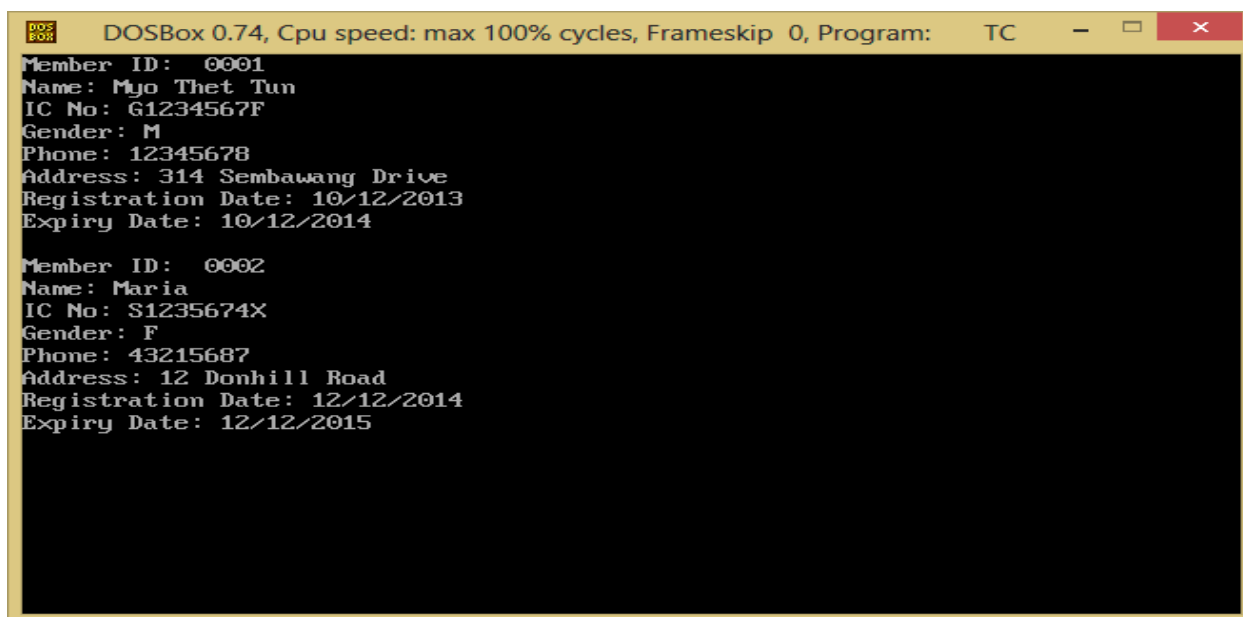
Test Case	18
Objectives	To check whether it shows the Display Member Details function if the user enters 2 in main menu choice.
Test Data	Enter '2'
Expected Test Result	If the user enters 2, all available member details will be displayed.
Actual Test Result	As shown below
Conclusion	Successful. All available member details were shown.
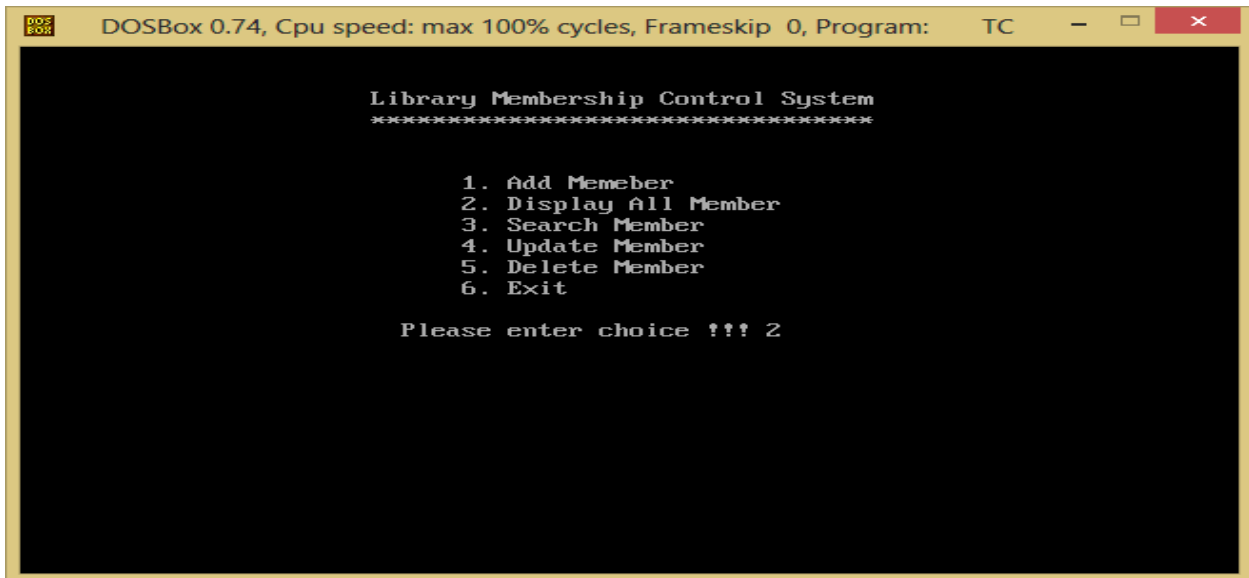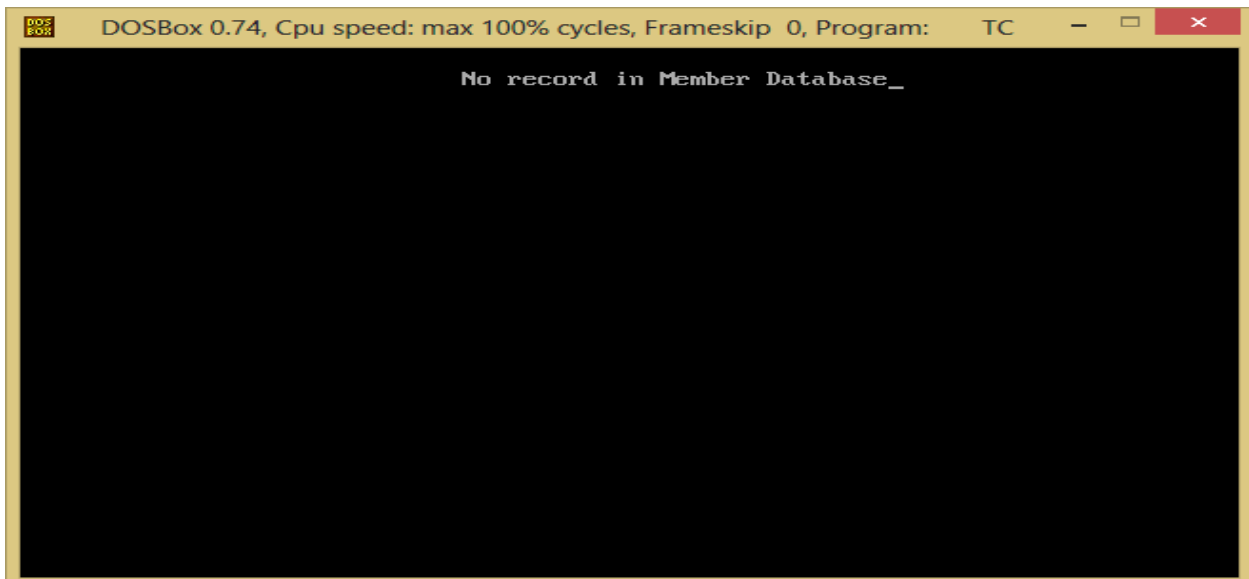
Test Case	19
Objectives	To check whether it shows error message if there is no record in the file when the user press 2 in main menu choice.
Test Data	Enter '2'
Expected Test Result	If the user enters '2', 'No record in database' error message will be displayed.
Actual Test Result	As shown below
Conclusion	Successful. Error message was shown.
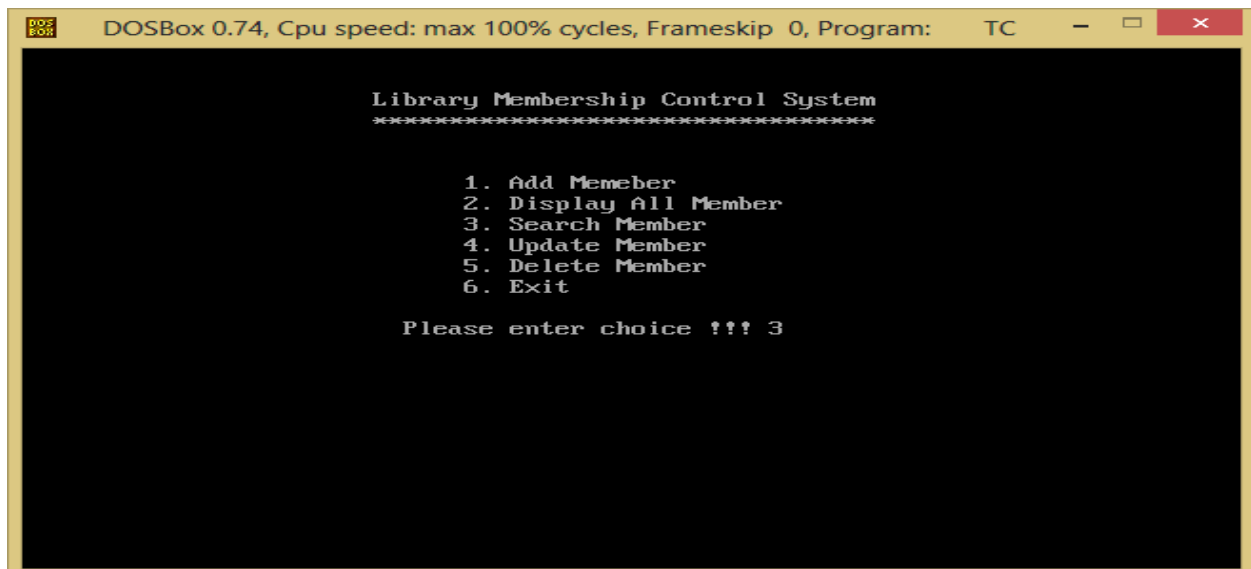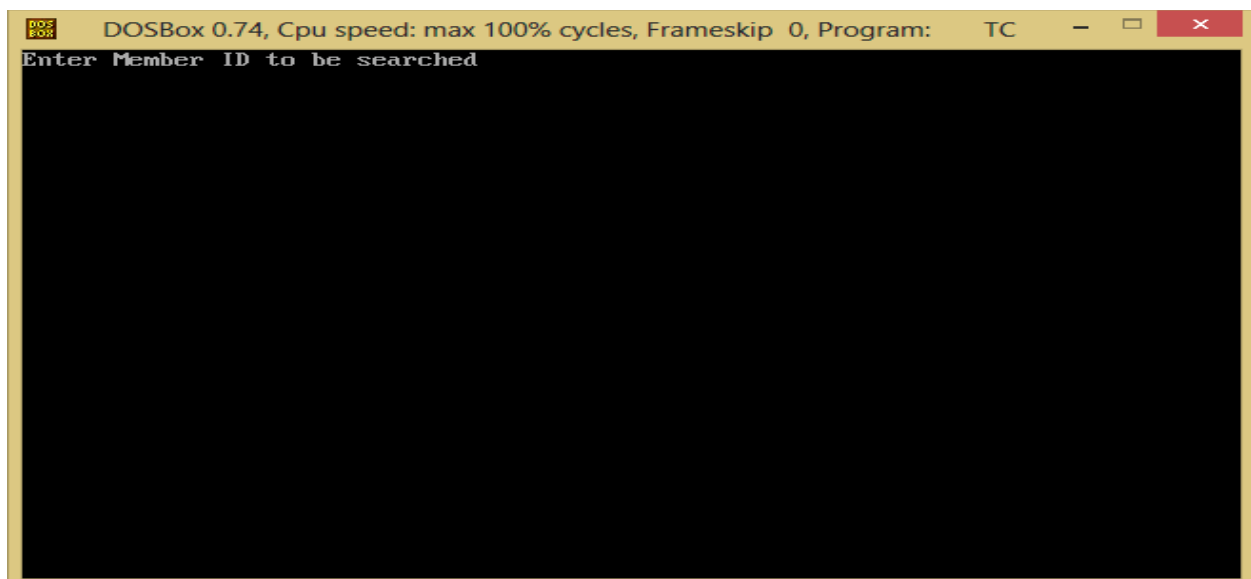
Test Case	20
Objectives	To check whether it shows the Search Member function if the user enters 3 in main menu choice.
Test Data	Enter '3'
Expected Test Result	If the user enters 3, member ID will be requested to enter.
Actual Test Result	As shown below
Conclusion	Successful. 'Enter member ID to be searched' was prompted.
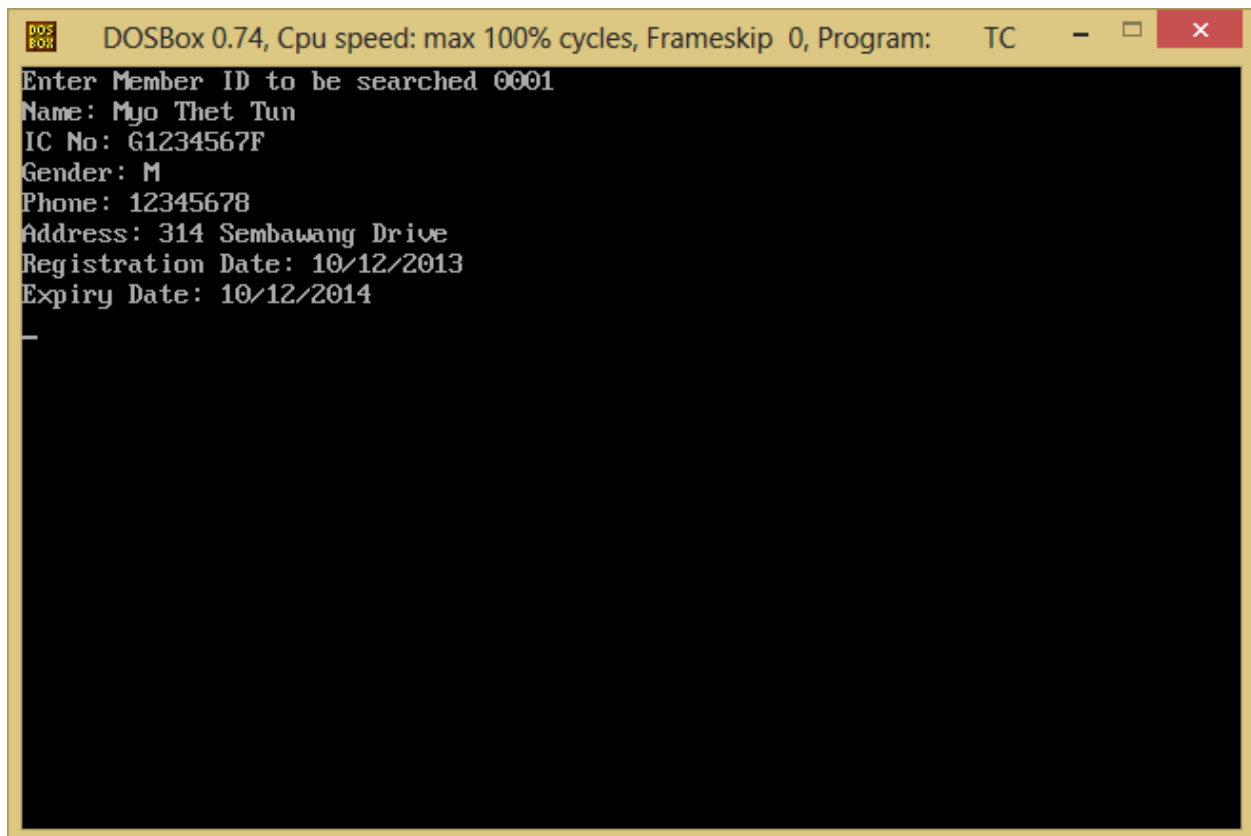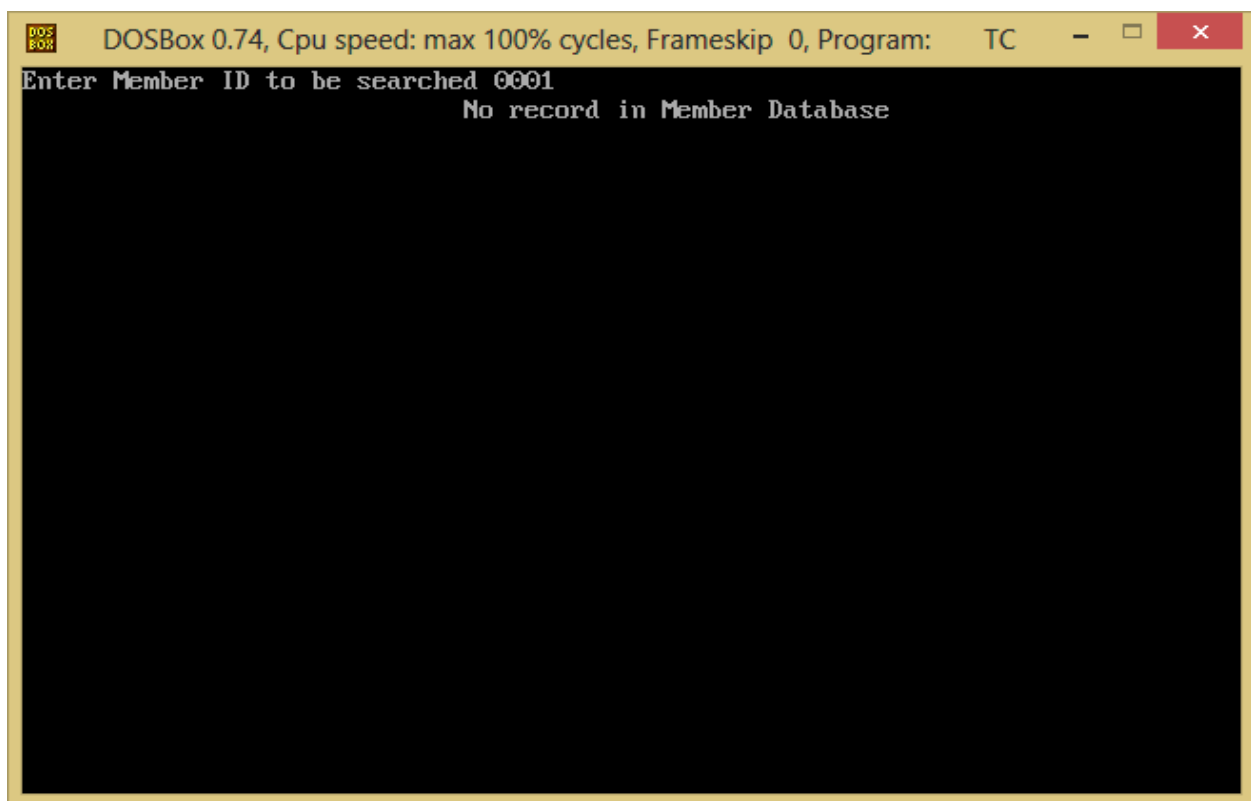
Test Case	21
Objectives	To check whether it shows member details if the user enters available ID in search member function.
Test Data	Enter '0001'
Expected Test Result	If the user enters 0001, member details will be shown.
Actual Test Result	As shown below
Conclusion	Successful. The details of member ID 0001 was displayed.
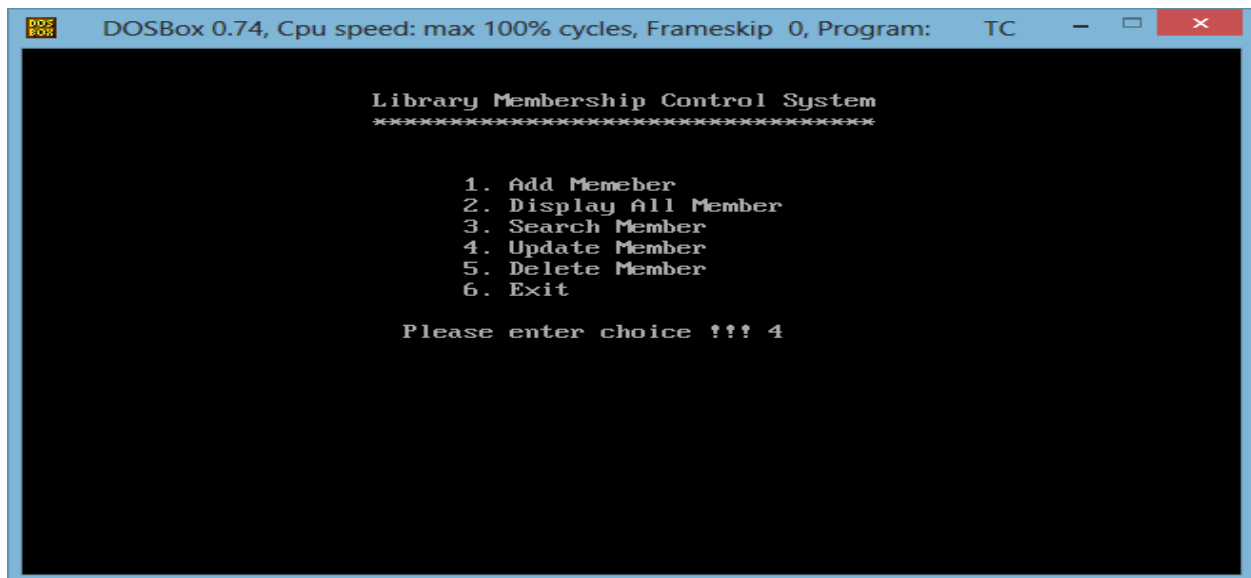
Test Case	22
Objectives	To check whether it shows error message if there is no record in the file when the user enters member ID to be searched.
Test Data	Enter '0001'
Expected Test Result	If the user enters '0001', 'No record in database' error message will be displayed.
Actual Test Result	As shown below
Conclusion	Successful. Error message was shown.
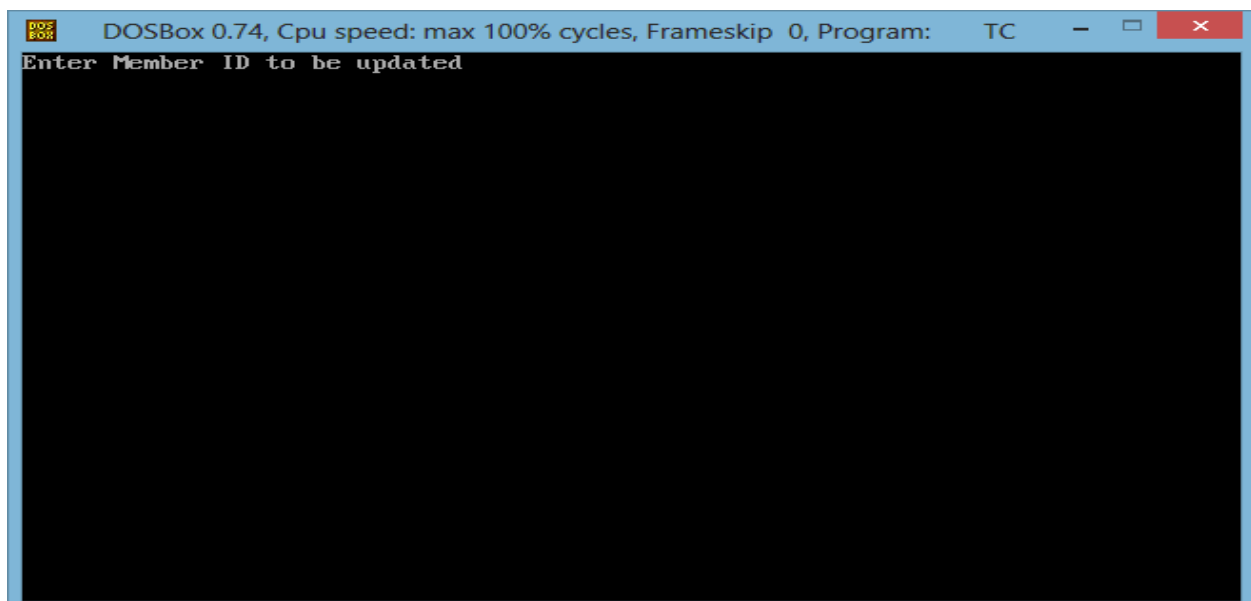
Test Case	23
Objectives	To check whether it shows the Update Member function if the user enters 4 in main menu choice.
Test Data	Enter '4'
Expected Test Result	If the user enters 4, member ID will be requested to enter.
Actual Test Result	As shown below
Conclusion	Successful. 'Enter member ID to be updated' was prompted.
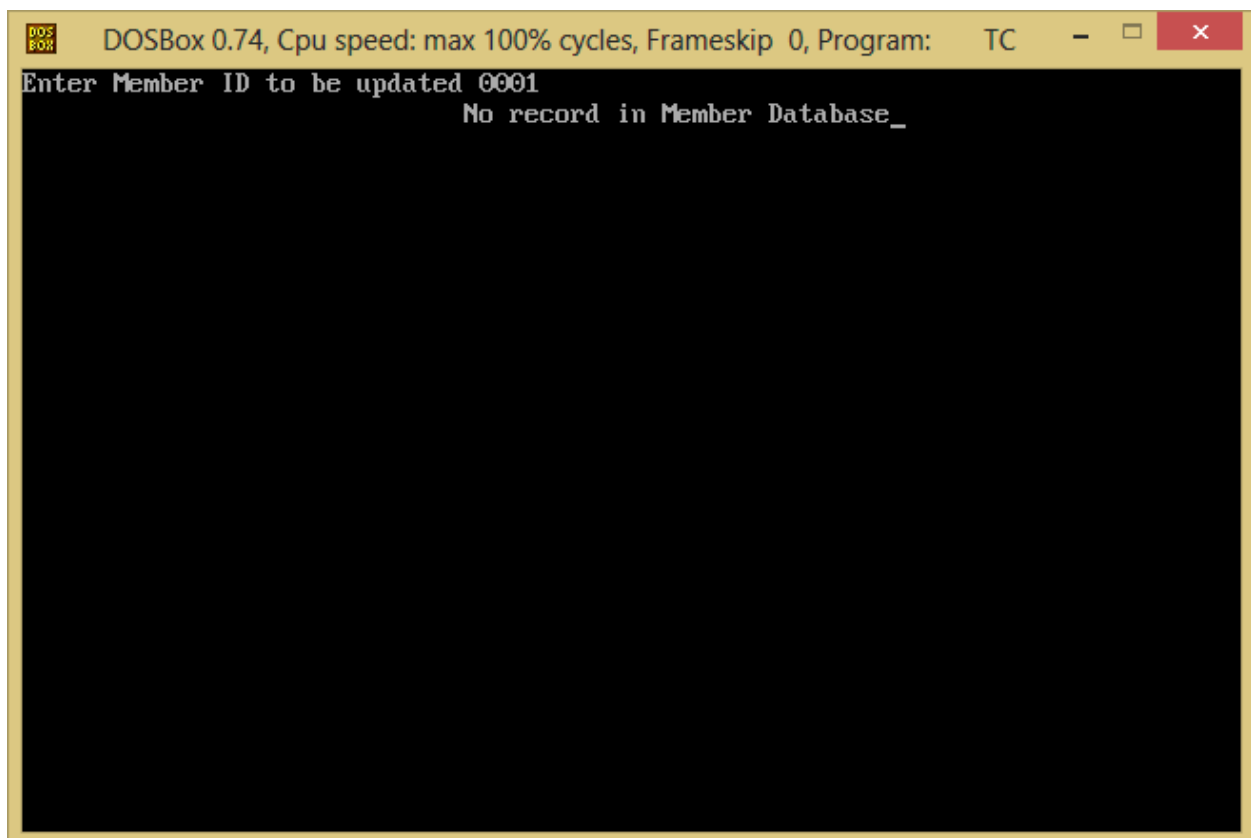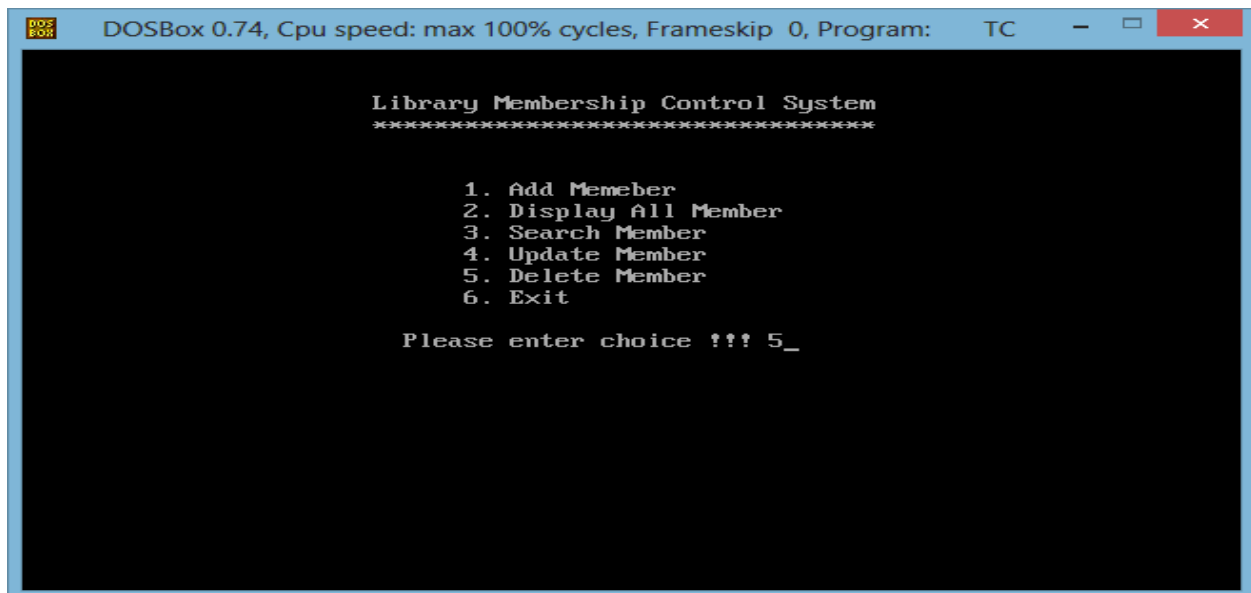
Test Case	24
Objectives	To check whether it shows error message if there is no record in the file when the user enters member ID to be updated.
Test Data	Enter '0001'
Expected Test Result	If the user enters '0001', 'No record in database' error message will be displayed.
Actual Test Result	As shown below
Conclusion	Successful. Error message was shown.

Test Case	25
Objectives	To check whether it shows the Delete Member function if the user enters 5 in main menu choice.
Test Data	Enter '5'
Expected Test Result	If the user enters 5, member ID will be requested to enter.
Actual Test Result	As shown below
Conclusion	Successful. 'Enter member ID to be deleted' was prompted.
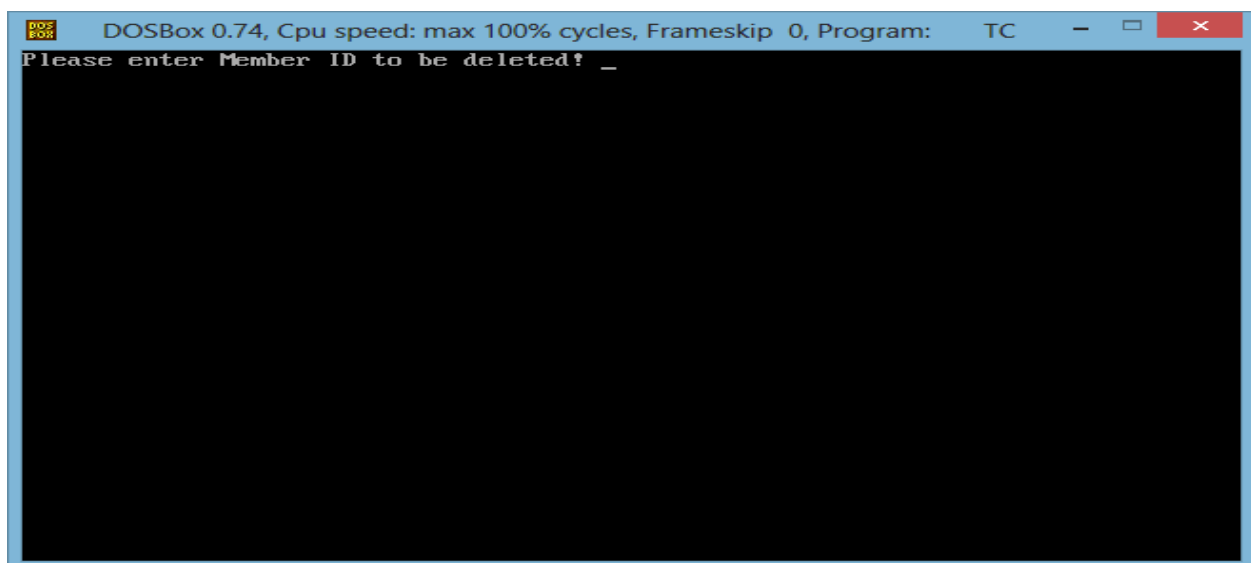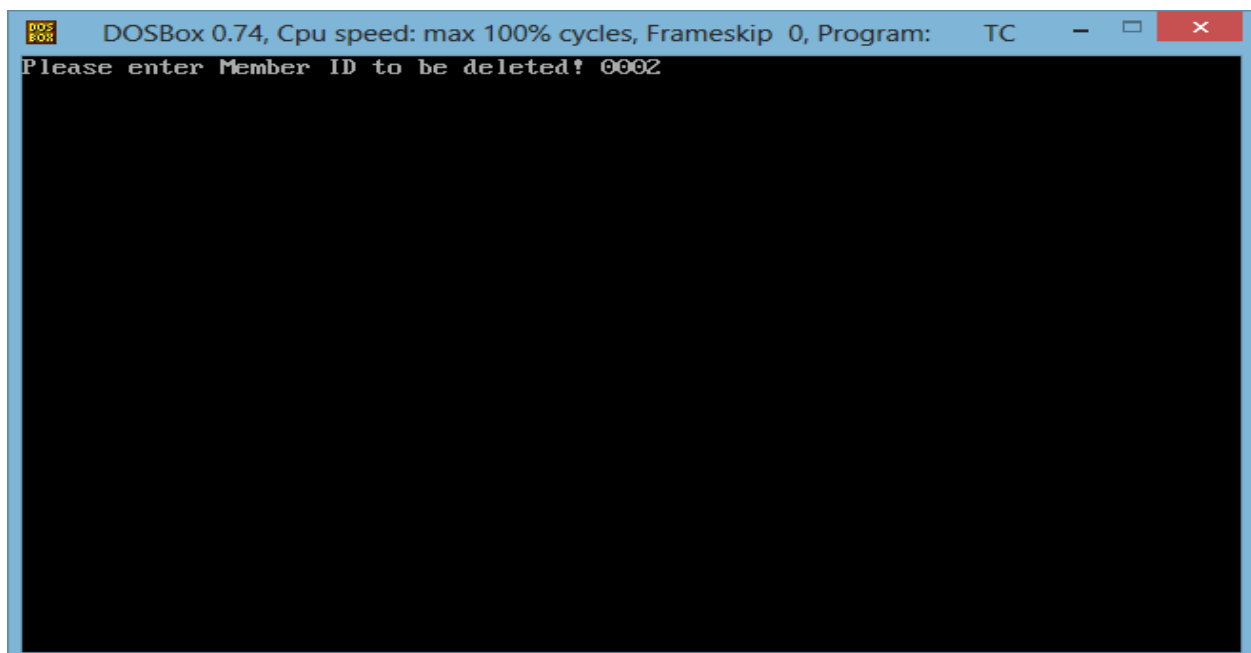
Test Case	26
Objectives	To check whether it shows error message if the user enters unavailable ID in delete member function.
Test Data	Enter '001'
Expected Test Result	If the user enters 001, error message will be shown.
Actual Test Result	As shown below
Conclusion	Successful. Error message was shown.

.
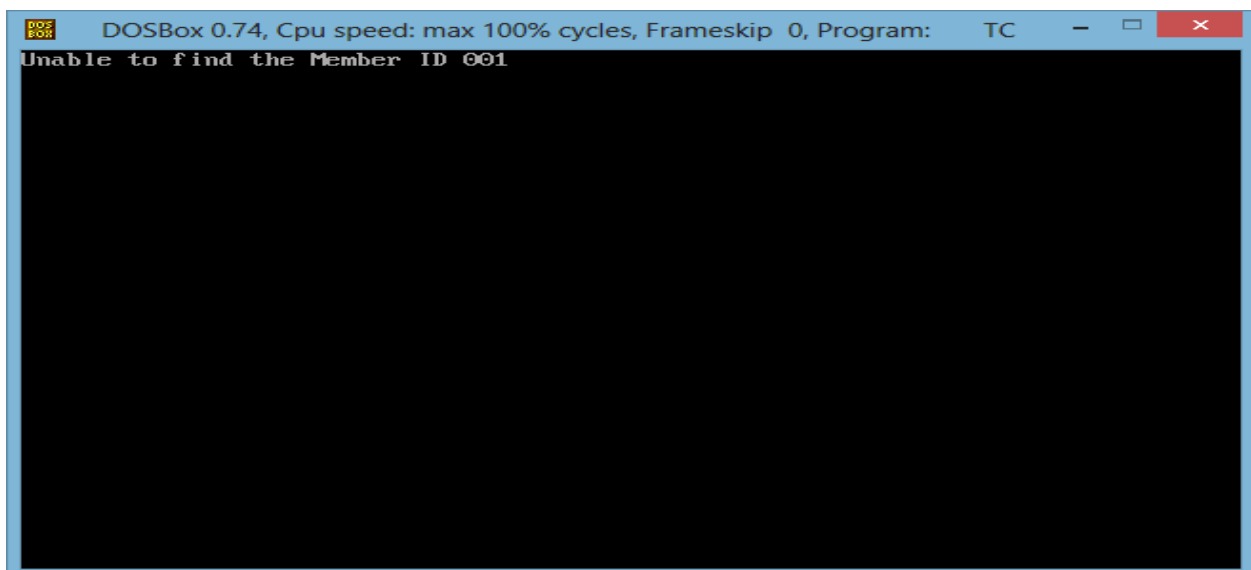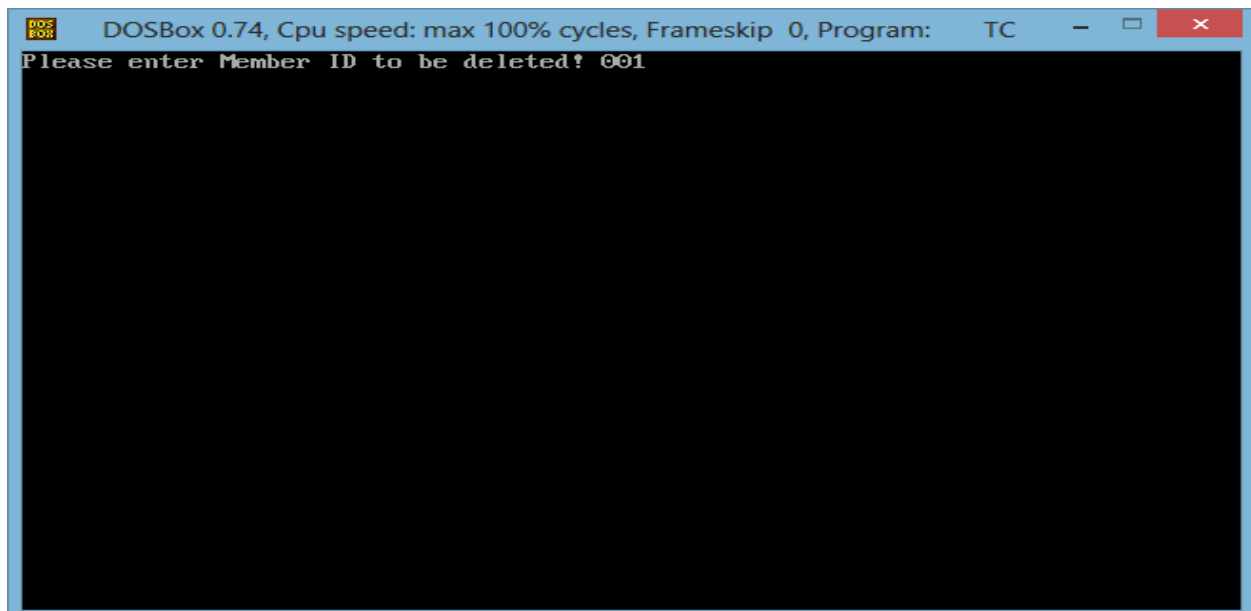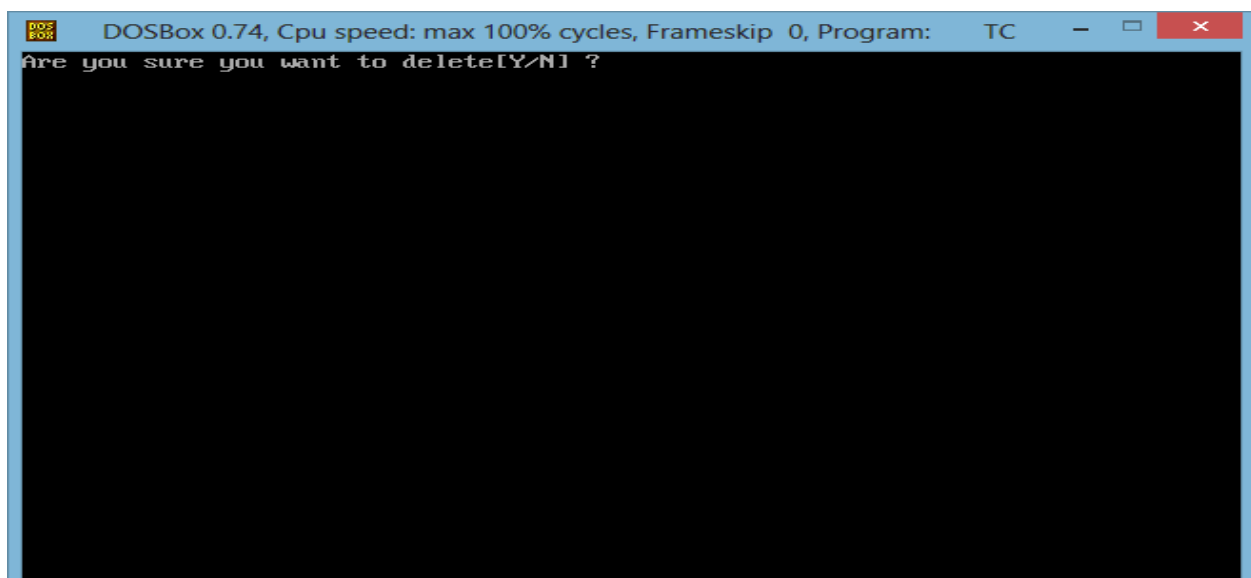
Test Case	27
Objectives	To check whether it shows confirm to be deleted message if the user enters available ID in delete member function.
Test Data	Enter '0002'
Expected Test Result	If the user enters 0002, 'Are you sure you want to delete?' will be shown.
Actual Test Result	As shown below
Conclusion	Successful. The message was shown.

4.2.3 Test Logs

Test Case	Result
Case 1	Successful
Case 2	Successful
Case 3	Successful
Case 4	Successful
Case 5	Successful
Case 6	Successful
Case 7	Successful
Case 8	Successful
Case 9	Successful
Case 10	Successful
Case 11	Successful
Case 12	Successful
Case 13	Successful
Case 14	Successful
Case 15	Successful
Case 16	Successful
Case 17	Successful
Case 18	Successful
Case 19	Successful
Case 20	Successful
Case 21	Successful
Case 22	Successful
Case 23	Successful
Case 24	Successful
Case 25	Successful
Case 26	Successful
Case 27	Successful

5. IMPLEMENTATION

5.1 Hardware and Software Requirements

Hardware requirements:

Intel Core i5-3230M Processor 2.60 GHz or higher

1 GB RAM for (32-bit) or 2 GB RAM for (64-bit)

Software requirements:

Windows Vista, 7 or 8 (32bit) or (64-bit)

Turbo C++ 3.0

CD-ROM Drive

5.2 User Manual
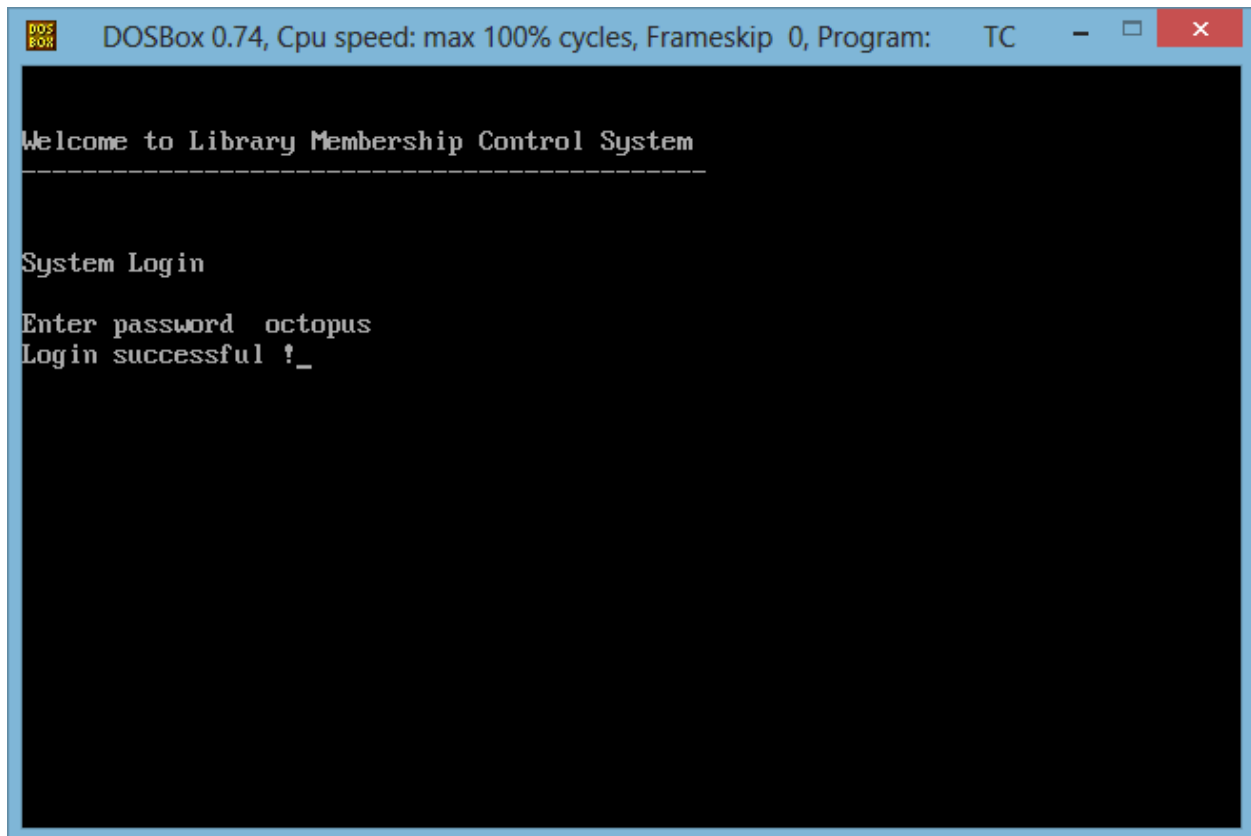
‘Welcome Screen’ appears.

Enter password ‘octopus’ for log in. System log in can be tried for 3 attempts if the user name or password is typed wrongly.
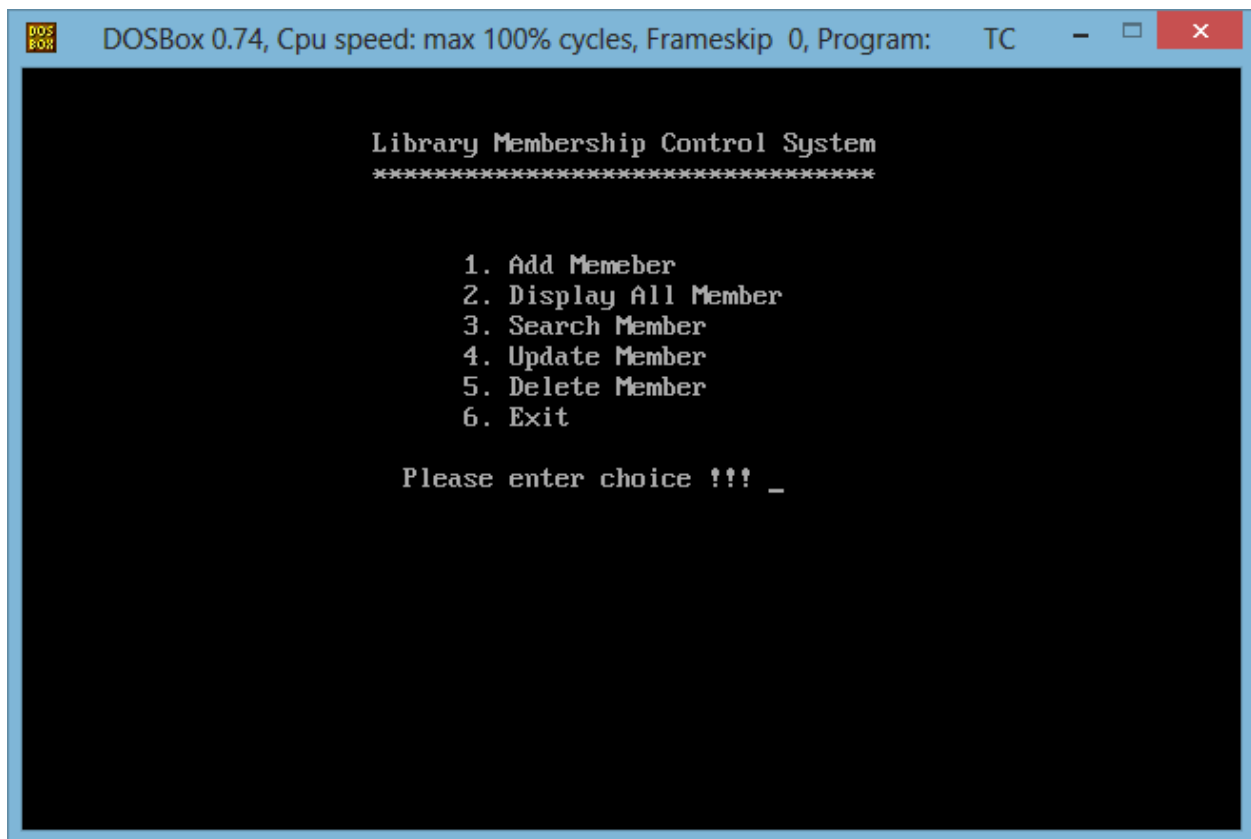
Main menu is displayed. Requested to enter for a choice.

Type 1 to add new member.

Type 2 to display all member details.
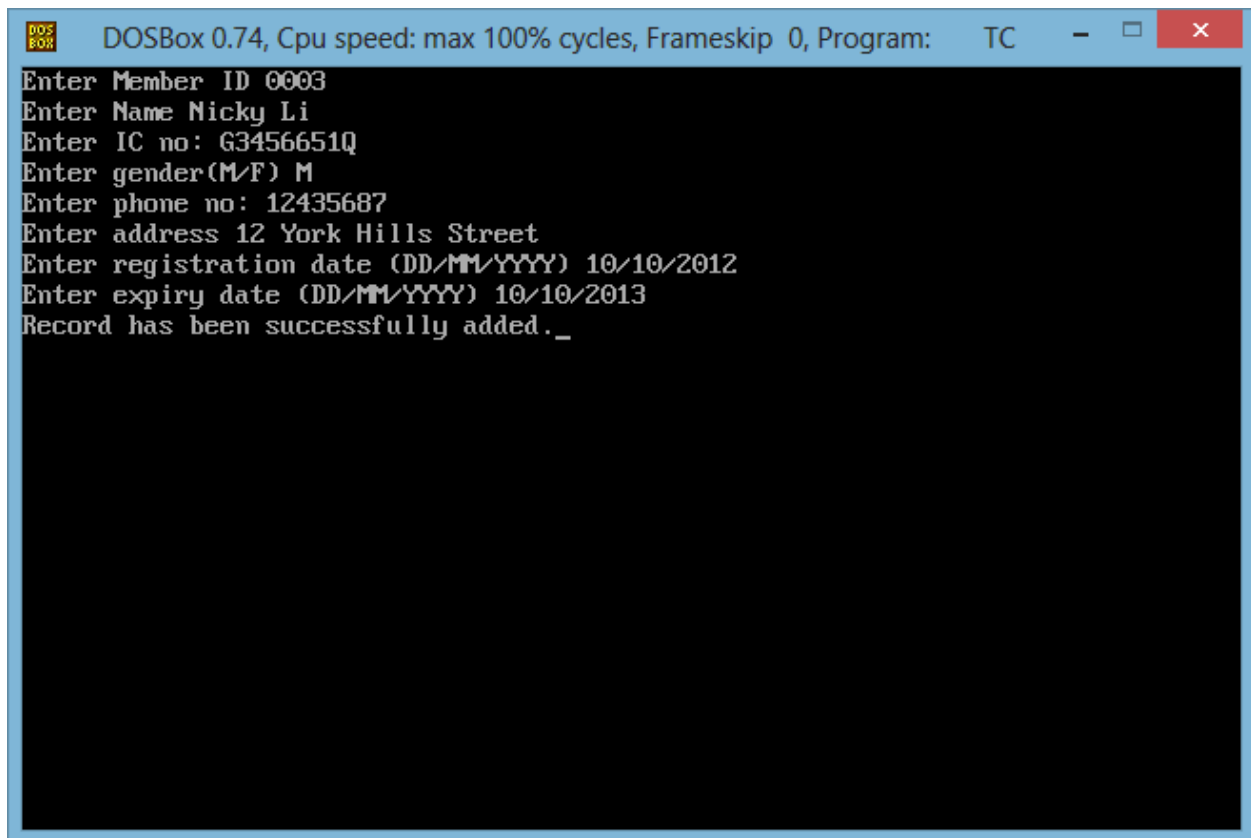
Type 3 to search for a specific member using Member ID.

Type 4 to update member details according to Member ID.

Type 5 to delete a member from the system using Member ID.

Type 6 to exit from the system.

Please type 1 in main menu to add a new member.

Member ID must have 4 digits only and cannot be duplicated.

Name has no limit character.

IC No. must have 9 characters.

Gender can be entered one character either (M/m) or (F/f).

Phone No. must have 8 characters.

Address has no limit character.

Registration Date must have 10 characters and must follow this date format (DD/MM/YYYY).

Expiry Date must have 10 characters and must follow this date format (DD/MM/YYYY).

Please type 2 in main menu to display all available members' details.

Currently, there are 3 members, Member ID 0001, 0002 and 0003, in the file. And their details will be displayed.
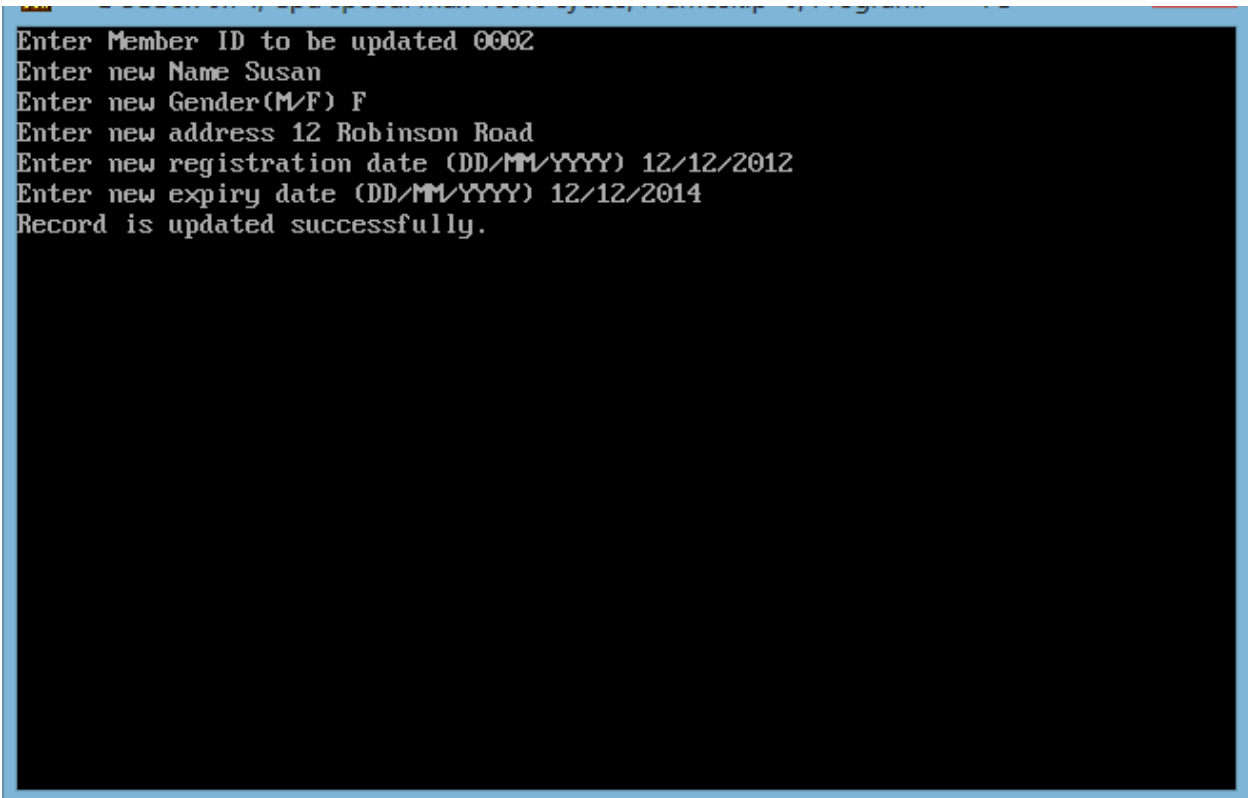
Please type 3 in main menu to search for a specific member's details using Member ID.

Enter member ID to be searched. Type member ID and the person's details will be displayed below. The user must type a valid ID format and must be an available ID.

Please type 4 to update member details according to Member ID.


```
Enter Member ID to be updated 0002
Enter new Name Susan
Enter new Gender(M/F) F
Enter new address 12 Robinson Road
Enter new registration date (DD/MM/YYYY) 12/12/2012
Enter new expiry date (DD/MM/YYYY) 12/12/2014
Record is updated successfully.
```

Enter member ID to be updated. User must enter a valid ID format and available ID in the file. If the user entered Member ID to be updated, it will prompt for new name, new IC No., new gender, new phone no. , new address, new registration date and new expiry date and all the new information will be displayed below as the user enter. The user must follow the same format as in Add New Member function as shown below when updating the record.

Member ID must have 4 digits only and cannot be duplicated.

Name has no limit character.

IC No. must have 9 characters.

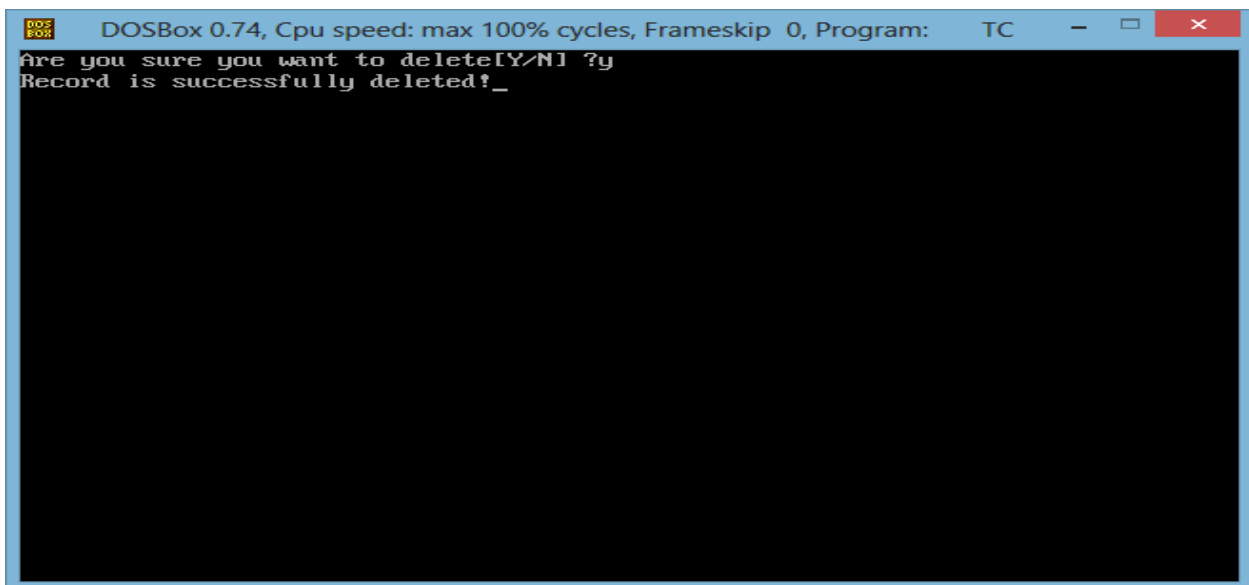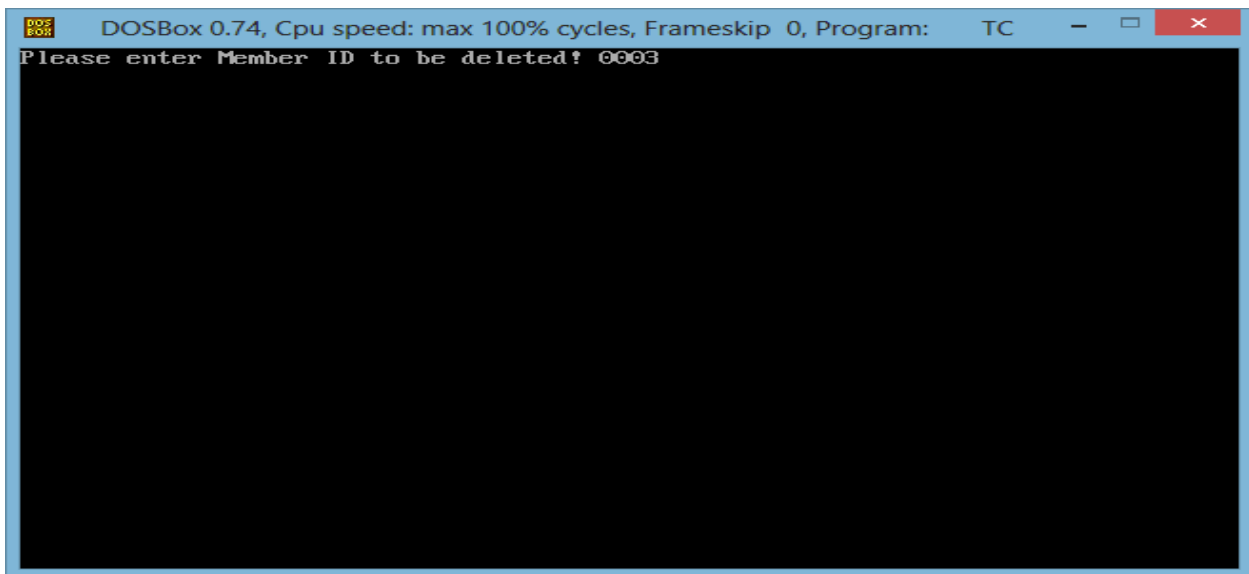Gender can be entered one character either (M/m) or (F/f).

Phone No. must have 8 characters.

Address has no limit character.

Registration Date must have 10 characters and must follow this date format (DD/MM/YYYY).
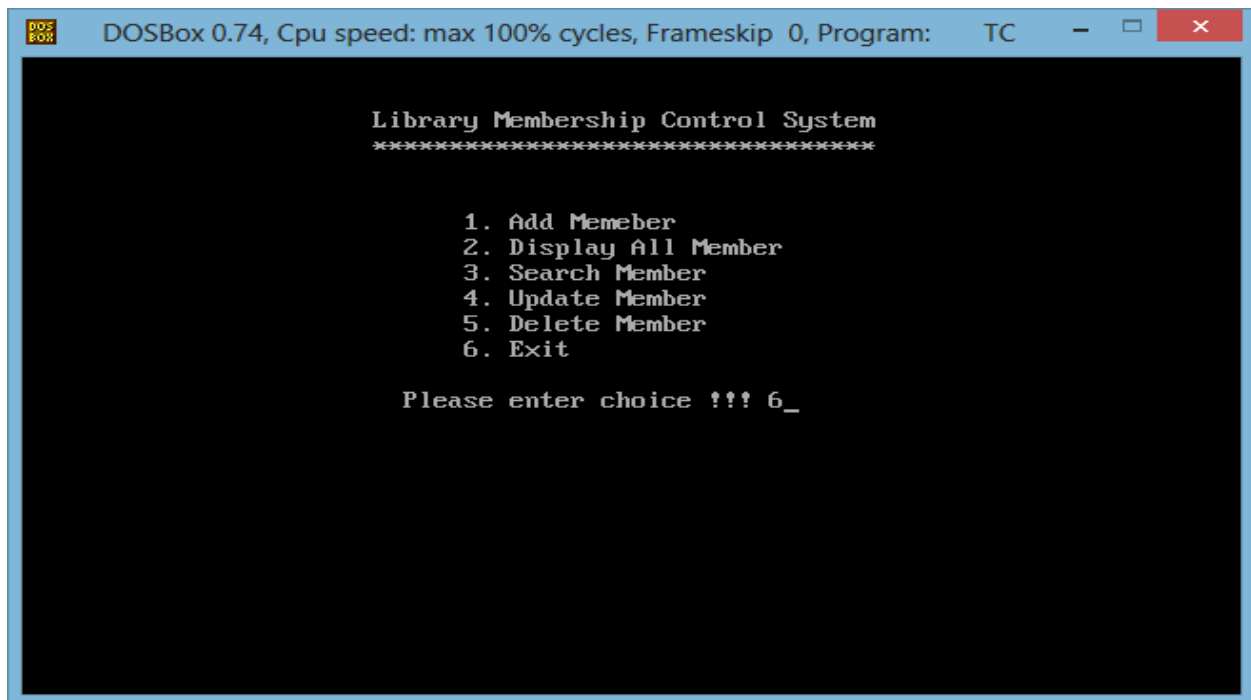
Expiry Date must have 10 characters and must follow this date format (DD/MM/YYYY).

Please type 5 to delete a member from the system using Member ID.

Enter Member ID to be deleted. If the user enters a valid and available ID, it will be deleted and 'The record has been deleted successfully' message will be appeared.

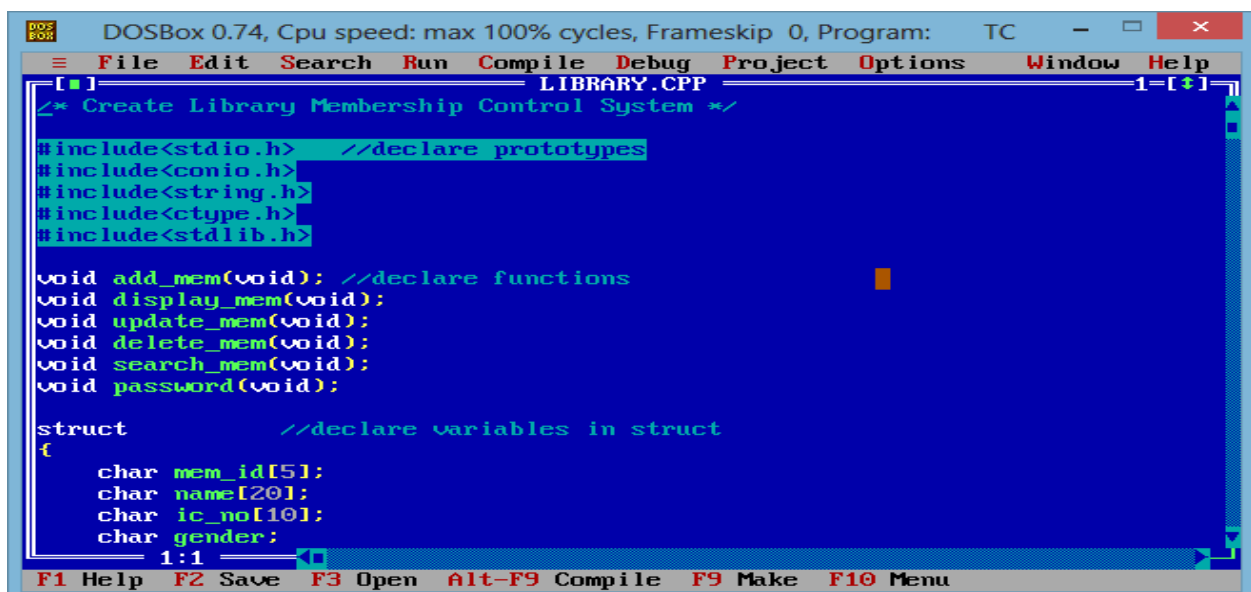Please type 6 to exit from the system.

If the user enters 6, the system will be exited.

6. CONCLUSION

6.1 Program Strengths

It is a security enabled system with many data validation checks such as fixed limit characters restrictions and maximum characters limits and date format restrictions and data duplication check for ID field and controlled by many error and information messages to guide users to type in valid data information.

6.2 Program Weaknesses

The main weakness of the program is that it cannot validate whether the user types in an alphabetic or a digit. And the password is set by the system and they cannot be changed.

6.3 Program Enhancements

In the future version of the program, validating checks for alphabetic or digit and change password function will be included and some functions will be expected to be improved.

****