2014

PROJECT TITLE:

Library Membership Control System

STUDENT NAME: Myo Thet Tun

STUDENT ID: 018800021524

SUBJECT CODE: IT209

CLASS CODE: P12May2014IT209

LECTURER NAME: Tan Kia Wa

SUBJECT: Java Programming

SUBMISSION DATE: 31 July 2014

TABLE OF CONTENTS

1. Analysis

1.1 Library's Background ---------- 3

1.2 Current System ---------- 3

1.3 Problems Background ---------- 4

1.4 Objectives ---------- 5

1.5 Record Database ---------- 6

2. Program Design

2.1 Flow Chart ---------- 8

2.2 Input Screen Design ---------- 11

3. Development and Testing

3.1 Program Code ---------- 12

3.2 Program Testing ---------- 26

Test Plan ---------- 26

Test Cases and Results ---------- 28

Test Logs ---------- 53

4. Implementation

4.1 Hardware and Software Requirements ---------- 54

4.2 User Manual ---------- 55

5. Conclusion

5.1 Program Strengths ---------- 62

5.2 Program Weaknesses ---------- 62

5.3 Program Enhancements ---------- 62

1. ANALYSIS

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 Problems background

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.

1.4 Objectives

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

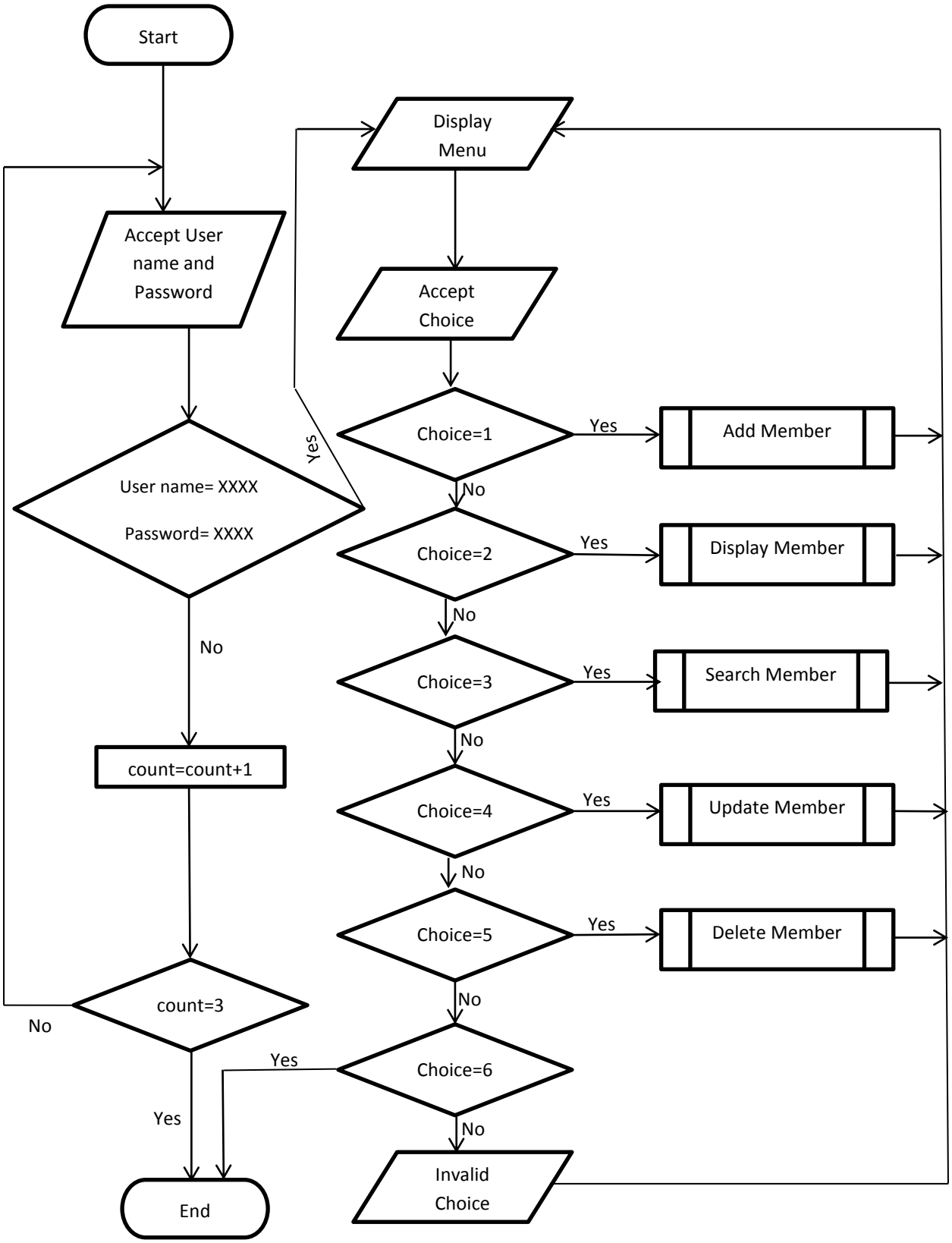
- Delete record using Member ID

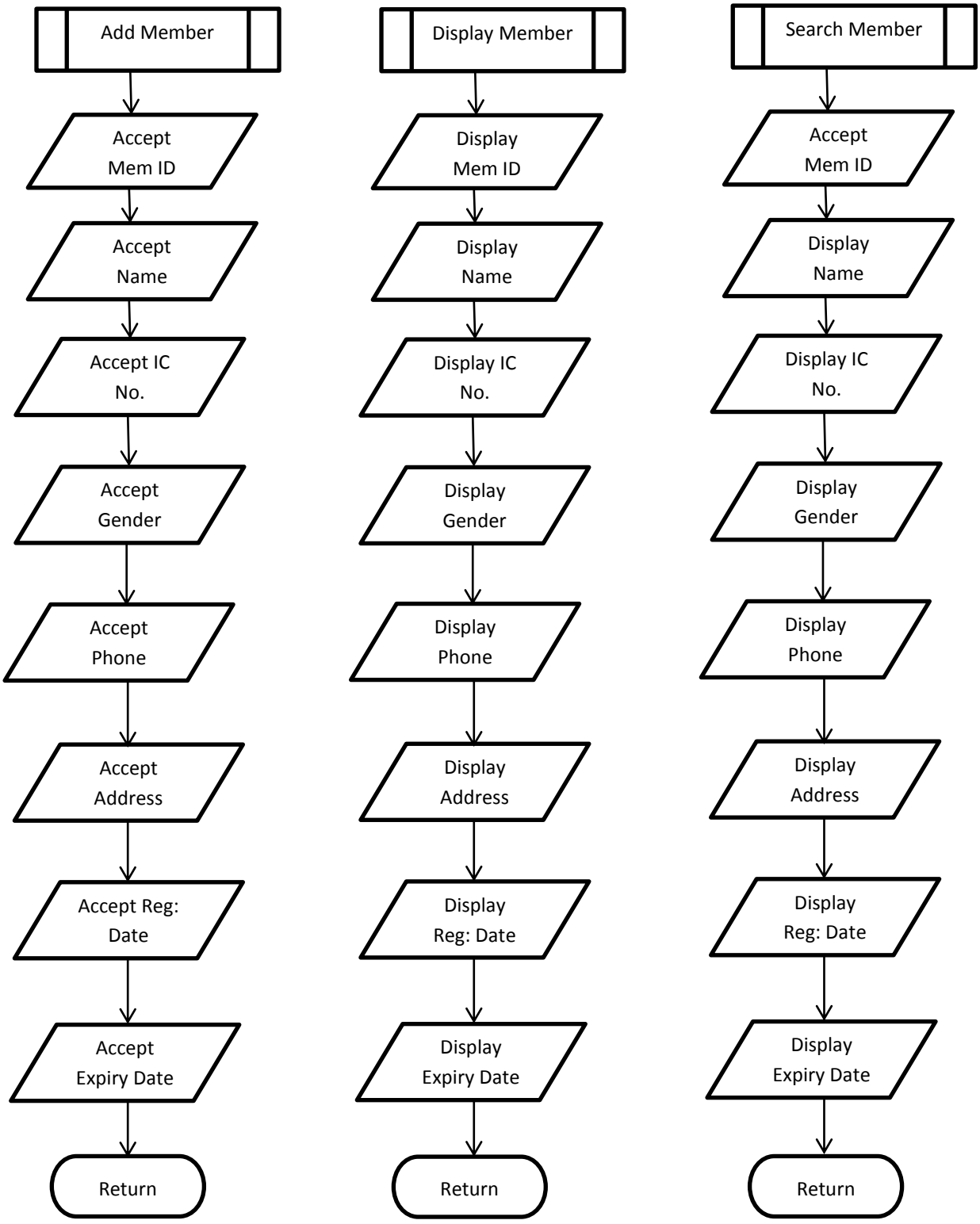
6. Exit

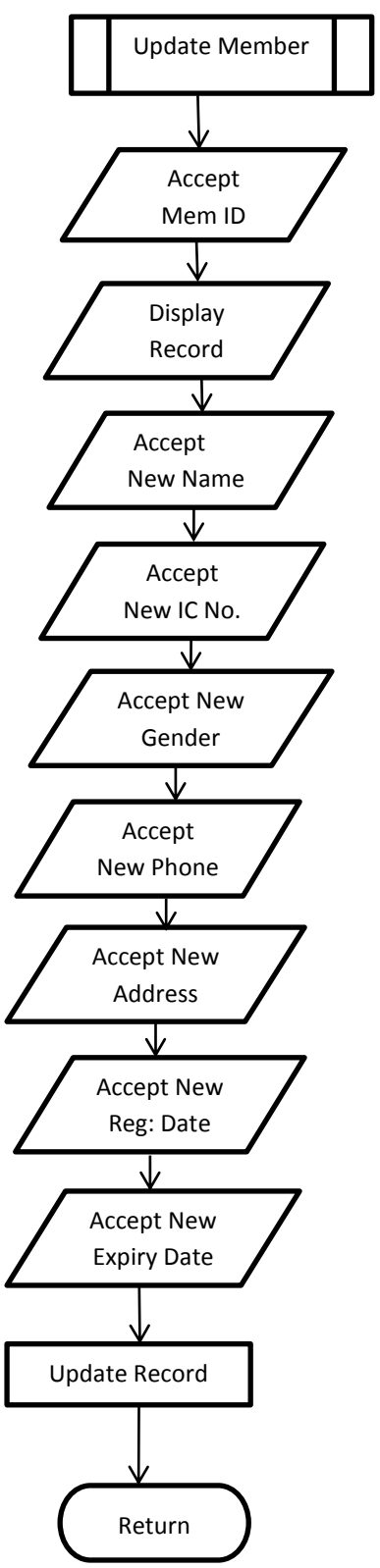
- Exit from the program

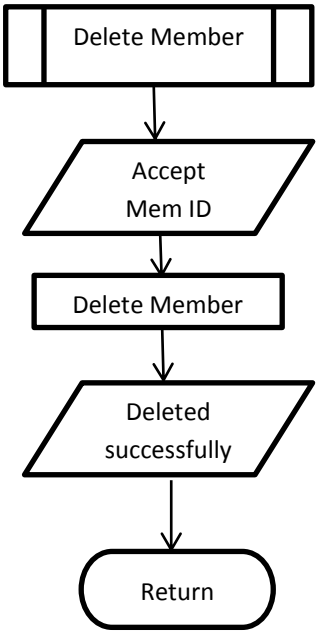
2. PROGRAM DESIGN

2.1 Flow Chart

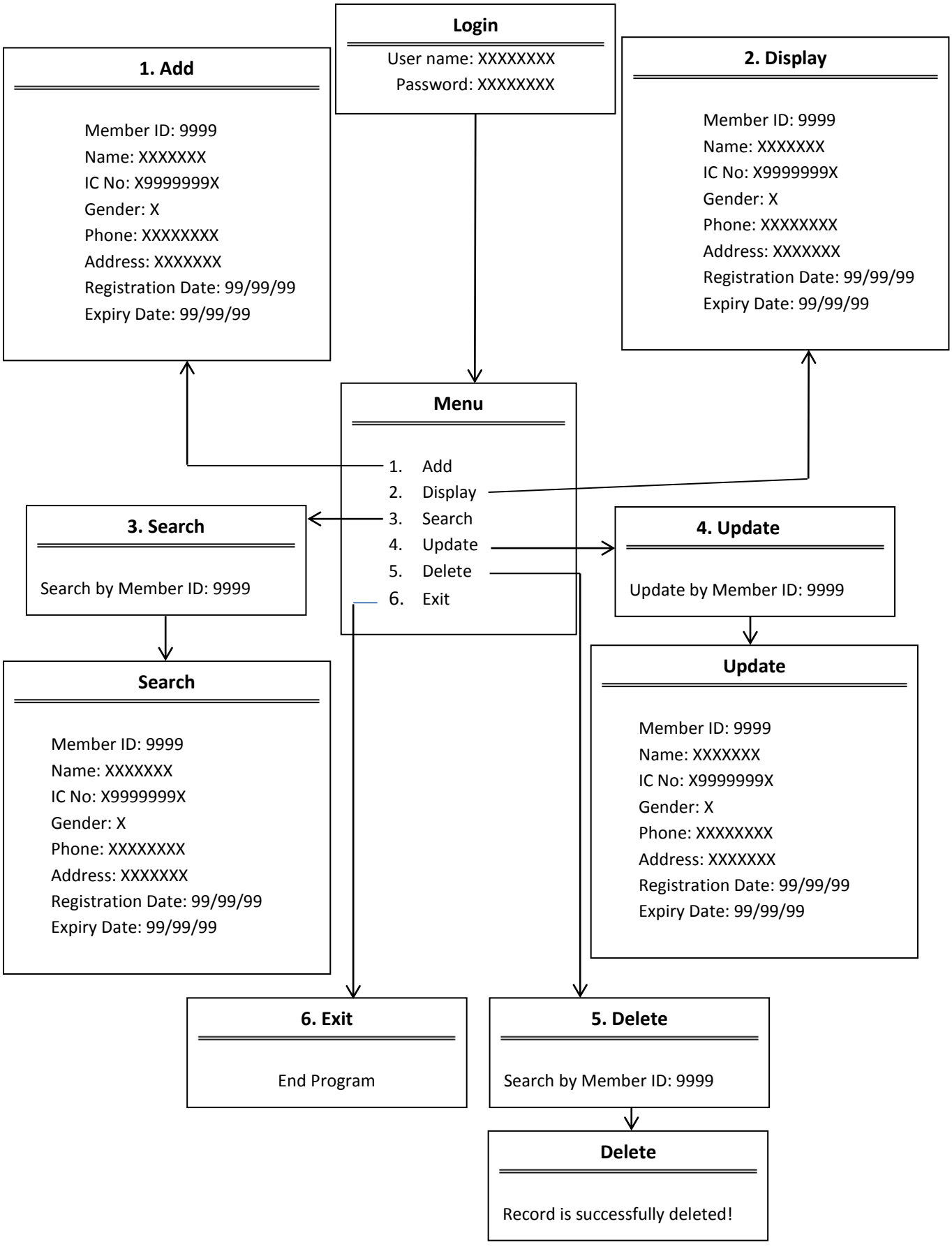



Add Member
Accept Mem ID
Accept Name
Accept IC No.
Accept Gender
Accept Phone
Accept Address
Accept Reg: Date
Accept Expiry Date
Return
Display Member
Display Mem ID
Display Name
Display IC No.
Display Gender
Display Phone
Display Address
Display Reg: Date
Display Expiry Date
Return
Search Member
Accept Mem ID
Display Name
Display IC No.
Display Gender
Display Phone
Display Address
Display Reg: Date
Display Expiry Date
Return


Update Member
Accept
Mem ID
Display
Record
Accept
New Name
Accept
New IC No.
Accept New
Gender
Accept
New Phone
Accept New
Address
Accept New
Reg: Date
Accept New
Expiry Date
Update Record
Return


Delete Member
Accept
Mem ID
Delete Member
Deleted
successfully
Return

2.2 Input Screen Design



3. DEVELOPMENT AND TESTING

3.1 Program Code

setup.java

```
import java.io.*;

public class setup

{


public static void main(String[] args) throws Exception
                                        {
                                            RandomAccessFile data = new
                                        RandomAccessFile("library.dat","rw");
                                            data.writeInt(0);
                                            data.close();
                                        };
}
```

libra.java

```
import java.io.*;

public class libra

{

static BufferedReader input = new BufferedReader(new InputStreamReader(System.in));


static String mid[ ] = new String [9999];

static String name[ ] = new String[9999];

static String ic_no[ ] = new String[9999];

static char gender[ ] = new char[9999];

static String phone[ ] = new String[9999];
```

```
static String addr[ ] = new String[9999];
static String reg_date[ ] = new String[9999];
static String exp_date[ ] = new String[9999];

static int counter = 0;

public static void read_member( )throws Exception
{
        RandomAccessFile data = new RandomAccessFile("library.dat","rw");
        counter = data.readInt( );

        for(int x=0; x < counter; x++)
        {
                mid[x] = data.readUTF( );
                name[x] = data.readUTF( );
                ic_no[x] = data.readUTF( );
                gender[x] = data.readChar( );
                phone[x] = data.readUTF( );
                addr[x] = data.readUTF( );
                reg_date[x] = data.readUTF( );
                exp_date[x] = data.readUTF( );
        };
        data.close( );
};

public static void write_to_file( )throws Exception
{
```

```
        RandomAccessFile data = new RandomAccessFile("library.dat","rw");
        data.writeInt(counter);

        for(int x=0; x < counter; x++)
        {
                data.writeUTF(mid[x]);
                data.writeUTF(name[x]);
                data.writeUTF(ic_no[x]);
                data.writeChar(gender[x]);
                data.writeUTF(phone[x]);
                data.writeUTF(addr[x]);
                data.writeUTF(reg_date[x]);
                data.writeUTF(exp_date[x]);
        };
        data.close( );
};

public static void welcome( )
{
        System.out.println("\n\n\t\tWelcome to Library Membership Control System");
        System.out.println("\t\t****************************************");
}

public static void main(String[]args) throws Exception
{
        read_member();
```

```
        welcome();

        int count=1;
        do{
                System.out.println("Please Enter User Name ");
                String username = input.readLine();
                System.out.println("Please Enter Password ");
                String password = input.readLine();

                if(username.equals("nicky") && password.equals("octopus"))
                {
                        show_menu( );
                        count=4;
                }

                else
                        System.out.println("Invalid Login !!!");

                count++;

        }while(count<=3);

        write_to_file();
}

public static void show_menu() throws Exception
{
```

```
int choice = 0;

do{
        System.out.println("\n\n\t\t     Library Membership Control System");
        System.out.println("\t\t     **********************************\n\n\n");
        System.out.println("\t\t\t  1. Add New Member");
        System.out.println("\t\t\t  2. Display Member Details");
        System.out.println("\t\t\t  3. Search Member Details");
        System.out.println("\t\t\t  4. Update Member Details");
        System.out.println("\t\t\t  5. Delete Member From System");
        System.out.println("\t\t\t  6. Exit From The System");
        System.out.println("\n\t\t\t  Please Enter A Choice : ");

        choice = Integer.parseInt(input.readLine());

        switch(choice)
        {
                case 1: add_member( );
                break;
                case 2: display_member( );
                break;
                case 3: search_member( );
                break;
                case 4: update_member( );
                break;
                case 5:  delete_member( );
                break;
```

```
                };
        }while(choice != 6);
}

public static void add_member( ) throws Exception
{
        for(int x = 0;x < counter; x++)
        {
                do{
                        System.out.println("Please Enter Member ID ");
                        mid[counter] = input.readLine();

                        if(mid[counter].equals(mid[x]) || mid[counter].length() !=4)
                        {
                                System.out.println("Invalid ID !!!");
                        };

                }while(mid[counter].equals(mid[x]) || mid[counter].length() !=4);
        };


        do{
                System.out.println("Please Enter Name ");
                name[counter] = input.readLine( );
```

```
            if(name[counter].length() > 40)
                    System.out.println("Name must not have more than 40 characters");

    }while(name[counter].length() > 40);

    do{
            System.out.println("Please Enter IC No. ");
            ic_no[counter] = input.readLine( );

            if(ic_no[counter].length() != 9)
                    System.out.println("IC No. must have 9 characters only");

    }while(ic_no[counter].length() != 9);

    System.out.println("Please Enter Gender(M/F) ");
    gender[counter] = Character.toUpperCase((char)System.in.read( ));
    System.in.read();System.in.read();

    do{
            System.out.println("Please Enter Phone No. ");
            phone[counter] = input.readLine( );

            if(phone[counter].length() != 8)
                    System.out.println("Phone No. must have 8 characters only");

    }while(phone[counter].length() != 8);
```

```
do{
        System.out.println("Please Enter Address ");
        addr[counter] = input.readLine( );

        if(addr[counter].length() > 60)
                System.out.println("Address must not have more than 60 characters");

}while(addr[counter].length() > 60);

do{
        System.out.println("Please Enter Registration Date(DD/MM/YY) ");
        reg_date[counter] = input.readLine( );

        if(reg_date[counter].length() != 8 || reg_date[counter].indexOf("/") ==-1)
                System.out.println("Registration Date must meet this format(DD/MM/YY)");

}while(reg_date[counter].length() != 8 || reg_date[counter].indexOf("/") ==-1);

do{
        System.out.println("Please Enter Expiry Date(DD/MM/YY) ");
        exp_date[counter] = input.readLine( );

        if(exp_date[counter].length() != 8 || exp_date[counter].indexOf("/") ==-1)
                System.out.println("Expiry Date must meet this format(DD/MM/YY)");

}while(exp_date[counter].length() != 8 || exp_date[counter].indexOf("/") ==-1);
```

```
        counter++;
}

public static void display_member( )
{
        for(int x = 0; x < counter; x++)
        {
                System.out.println("\nMember ID " + mid[x]);
                System.out.println("Name " + name[x]);
                System.out.println("IC No. " + ic_no[x]);
                System.out.println("Gender " + gender[x]);
                System.out.println("Phone " + phone[x]);
                System.out.println("Address " + addr[x]);
                System.out.println("Registration Date " + reg_date[x]);
                System.out.println("Expiry Date " + exp_date[x]);
        };
}

public static void search_member( ) throws Exception
{
        String temp_id;
        boolean found = false;

        System.out.println("Enter Member ID to be searched ");
        temp_id = input.readLine( );

        for(int x = 0; x < counter; x++)
```

```
        {
                if (temp_id.equals(mid[x]))
                {
                        found = true;

                        System.out.println("Member ID " + mid[x]);
                        System.out.println("Name " + name[x]);
                        System.out.println("IC No. " + ic_no[x]);
                        System.out.println("Gender " + gender[x]);
                        System.out.println("Phone " + phone[x]);
                        System.out.println("Address " + addr[x]);
                        System.out.println("Registration Date " + reg_date[x]);
                        System.out.println("Expiry Date " + exp_date[x]);
                };
        };
        if (found == false)
                System.out.println("Requested Member ID Not Found !!! ");
}

public static void delete_member( ) throws Exception
{
        String temp_id;
        boolean found = false;

        System.out.println("Enter Member ID to be deleted ");
        temp_id = input.readLine( );
```

```
	for(int x = 0; x < counter; x++)
	{
		if (temp_id.equals(mid[x]))
		{
			found = true;
			for(int y=x; y < counter;y++)
			{
				mid[y] = mid[y+1];
				name[y] = name[y+1];
				ic_no[y] = ic_no[y+1];
				gender[y] = gender[y+1];
				phone[y] = phone[y+1];
				addr[y] = addr[y+1];
				reg_date[y] = reg_date[y+1];
				exp_date[y] = exp_date[y+1];
			};
			counter--;
			System.out.println("The record has been deleted successfully !!!");
		};
	};
	if (found == false)
		System.out.println("Requested Member ID Not Found !!! ");
}

public static void update_member( ) throws Exception
{
	String temp_id;
```

```
boolean found = false;

System.out.println("Enter Member ID to be updated ");
temp_id = input.readLine( );

for(int x = 0; x < counter; x++)
{
        if (temp_id.equals(mid[x]))
        {
                found = true;

                System.out.println("Please Enter New Name ");
                String new_name = input.readLine( );

                if(new_name.length() > 40)
                        System.out.println("Name must not have more than 40 characters");

                name[x] = new_name;
                System.out.println("New Name is " + name[x]);

                System.out.println("Please Enter New IC No. ");
                String new_ic_no = input.readLine( );

                if(new_ic_no.length() !=9)
                        System.out.println("IC No. must have 9 characters only");

                ic_no[x] = new_ic_no;
```

```
                    System.out.println("New IC No. is " + ic_no[x]);

                    System.out.println("Please Enter New Gender(M/F) ");
                    char new_gender = Character.toUpperCase((char)System.in.read());
                    System.in.read();System.in.read();

                    gender[x] = new_gender;
                    System.out.println("New Gender is " + gender[x]);

                    System.out.println("Please Enter New Phone No. ");
                    String new_phone = input.readLine( );

                    if(new_phone.length() !=8)
                            System.out.println("Phone No. must have 8 characters only");

                    phone[x] = new_phone;
                    System.out.println("New Phone No. is " + phone[x]);

                    System.out.println("Please Enter New Address ");
                    String new_addr = input.readLine( );

                    if(new_addr.length() >60)
                            System.out.println("Address must not have more than 60
characters");

                    addr[x] = new_addr;
                    System.out.println("New Address is " + addr[x]);
```

```
                        System.out.println("Please Enter New Registration Date(DD/MM/YY) ");
                        String new_reg_date = input.readLine( );

                        if(new_reg_date.length() !=8 || new_reg_date.indexOf("/") ==-1)
                                System.out.println("Registration Date must meet this
format(DD/MM/YY)");

                        reg_date[x] = new_reg_date;
                        System.out.println("New Registration Date is " + reg_date[x]);

                        System.out.println("Please Enter New Expiry Date(DD/MM/YY) ");
                        String new_exp_date = input.readLine( );

                        if(new_exp_date.length() !=8 || new_exp_date.indexOf("/") ==-1)
                                System.out.println("Expiry Date must meet this
format(DD/MM/YY)");

                        exp_date[x] = new_exp_date;
                        System.out.println("New Expiry Date is " + exp_date[x]);

                };
        };

        if (found == false)
                System.out.println("Requested Member ID Not Found !!! ");
}
}
```

3.2 Program Testing

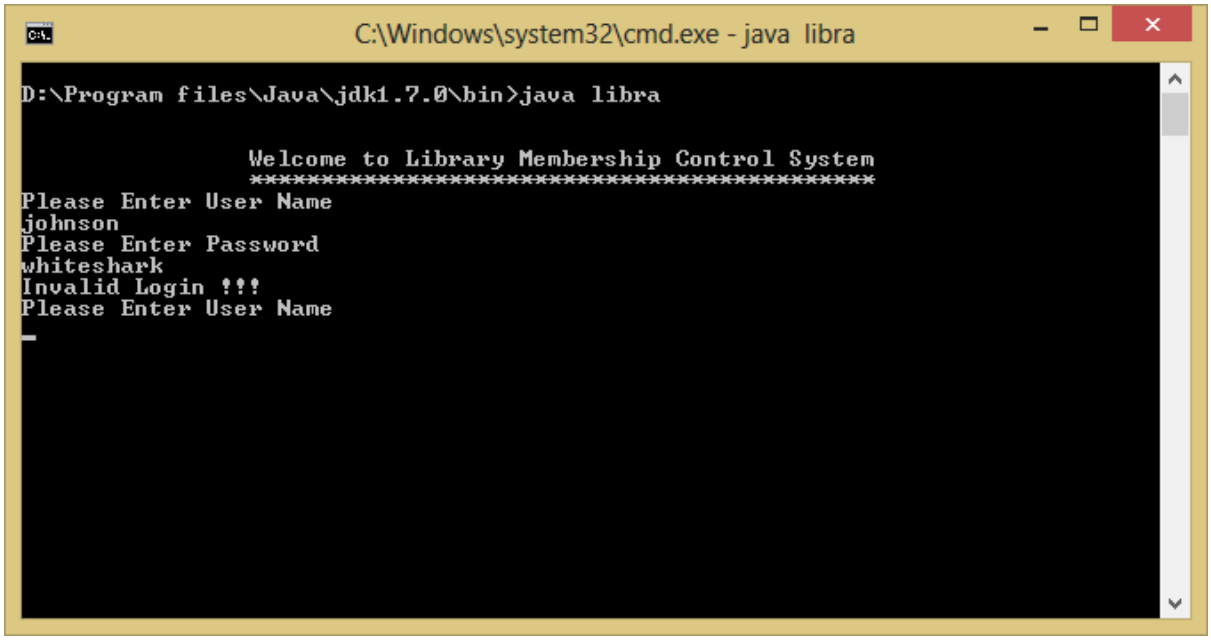
Test Plan

Test	Objectives
Case 1	To check whether it prompts up an error message 'Invalid Login' 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 the 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 shows error message if the user enters more than 4 characters in Member ID field.
Case 6	To check whether it shows error message if the user enters less than 4 characters in Member ID field.
Case 7	To check whether it prompt 'Enter name' if the user enters 4 characters in Member ID field.
Case 8	To check whether it prompt 'Enter IC No.' after the user enters name in Name field.
Case 9	To check whether it shows error message if the user enters more than 9 characters in IC No. field.
Case 10	To check whether it shows error message if the user enters less than 9 characters in IC No. field.
Case 11	To check whether it shows error message if the user enters invalid date format in Registration Date field.
Case 12	To check whether it prompt 'Enter Expiry Date' if the user enters valid date format in Registration Date field.
Case 13	To check whether it shows error message if the user enters invalid date format in Expiry Date field
Case 14	To check whether it shows Main Menu if the user enters valid date format in Expiry Date field.

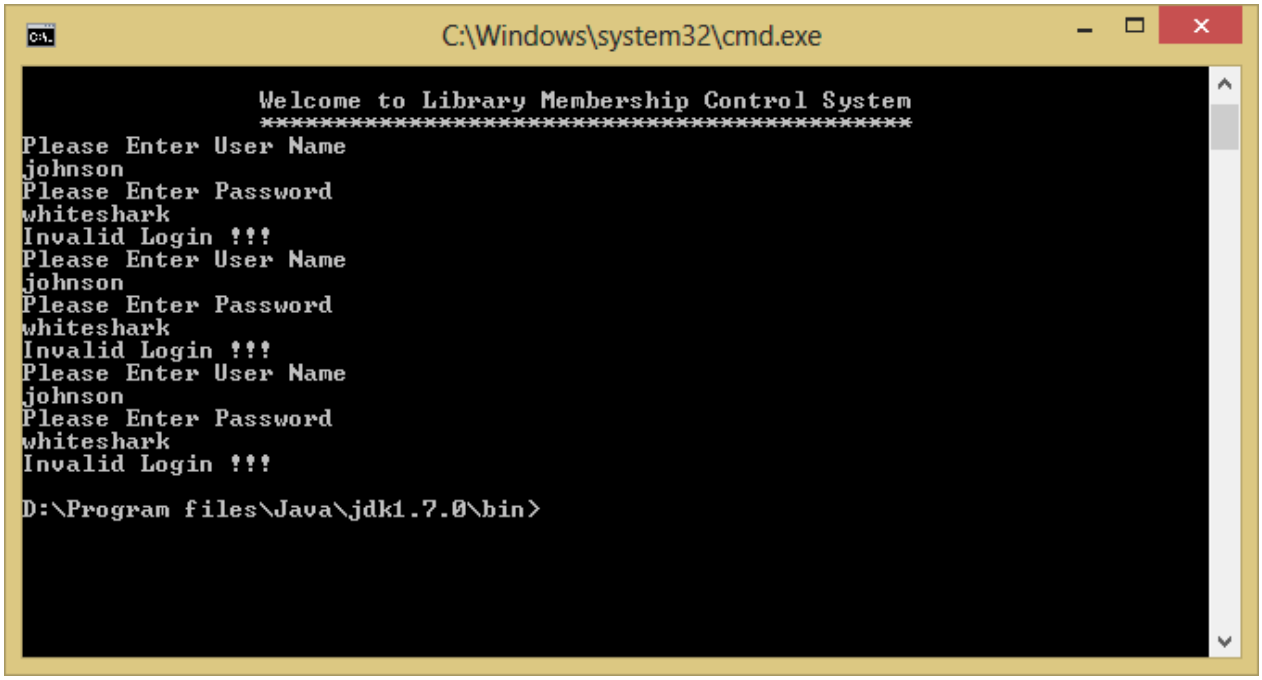
Case 15	To check whether it shows the Display Member Details function if the user enters 2 in main menu choice.
Case 16	To check whether it shows the Search Member Details function if the user enters 3 in main menu choice.
Case 17	To check whether it shows error message if the user enters unavailable ID in search member function.
Case 18	To check whether it shows member details if the user enters available ID in search member function.
Case 19	To check whether it shows the Update Member Details function if the user enters 4 in main menu choice.
Case 20	To check whether it shows error message if the user enters unavailable ID in update member function.
Case 21	To check whether it prompts to update member details if the user enters available ID in update member function.
Case 22	To check whether it shows the Delete Member function if the user enters 5 in main menu choice.
Case 23	To check whether it shows error message if the user enters unavailable ID in delete member function.
Case 24	To check whether it prompts to update member details if the user enters available ID in update member function.
Case 25	To check whether it exits from the system if the user enters 6 in main menu choice.

Test Cases and Results

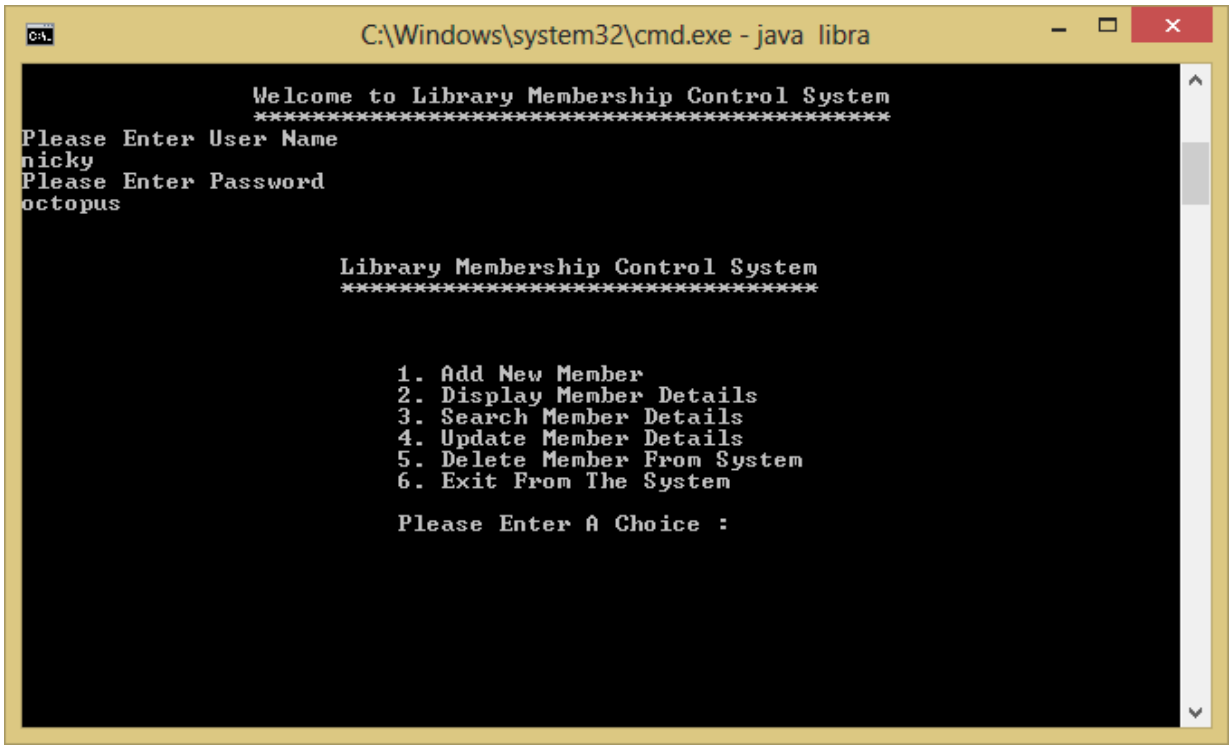
Test Case	1
Objectives	To check whether it prompts up an error message 'Invalid Login' when a user enters wrong username or password.
Test Data	'Enter User Name' & 'Enter Password'
Expected Test Result	If the wrong user name or password is entered, it will show 'Invalid Login' and prompt to enter for 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.



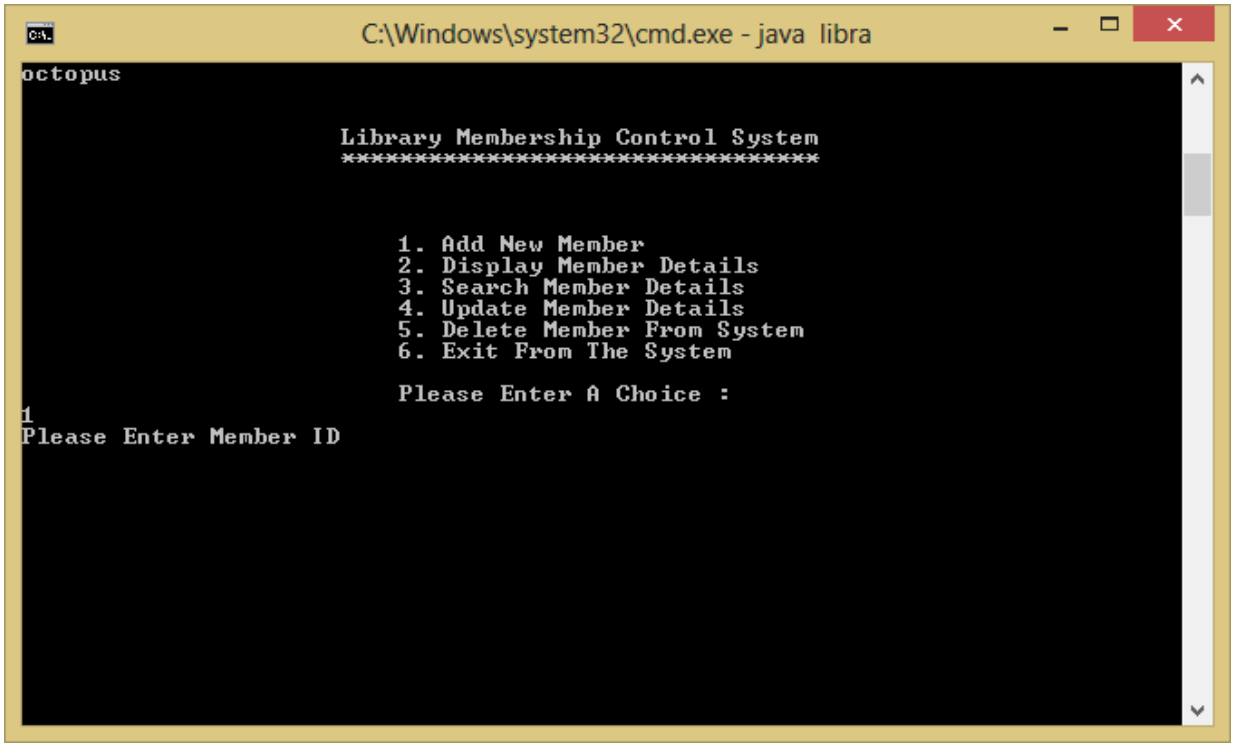
Test Case	2
Objectives	To check whether the system is totally shut down when user enters wrong username or password for 3 times.
Test Data	'Enter User Name' & 'Enter Password'
Expected Test Result	If the wrong user name or password is entered for 3 times, the system will be totally shut down.
Actual Test Result	As shown below
Conclusion	Successful. The system was totally shut down when the wrong user name or password is entered for 3 times.



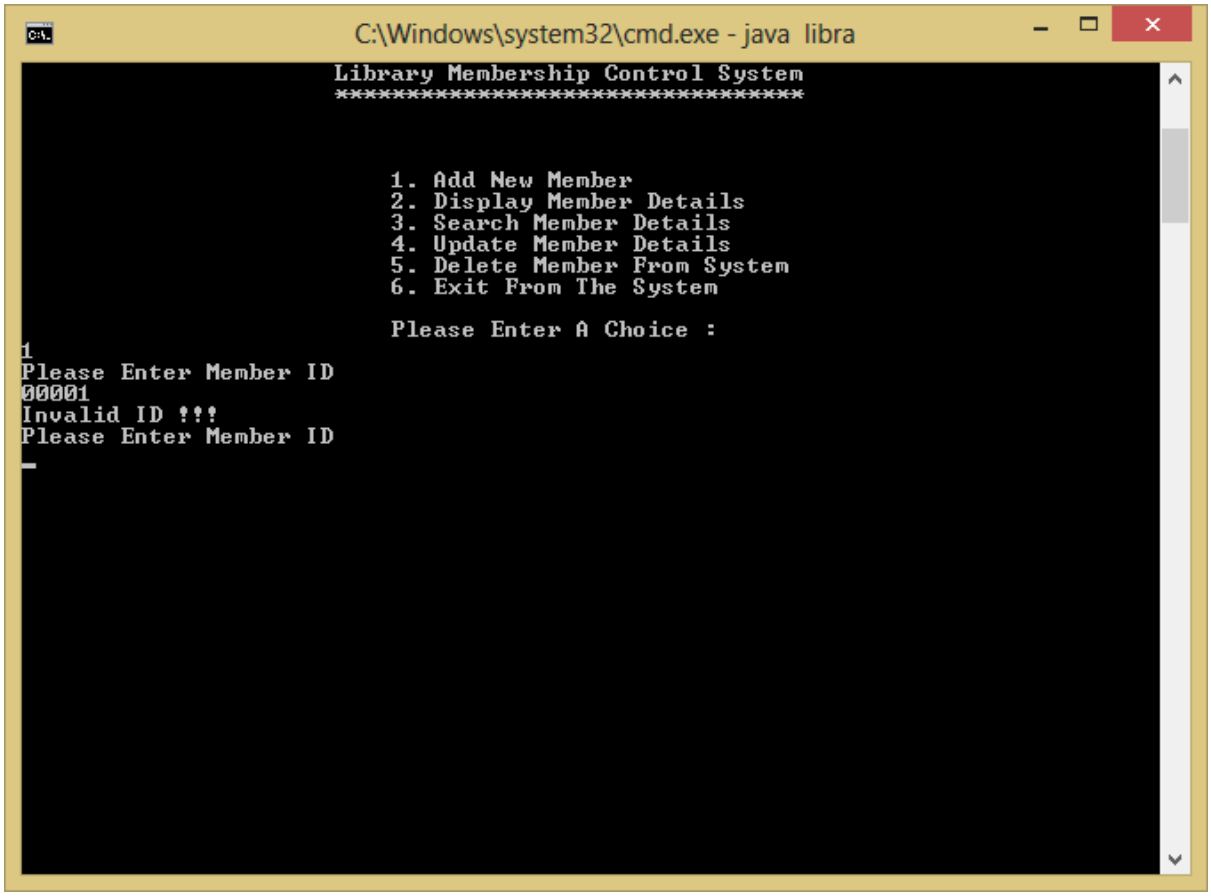
Test Case	3
Objectives	To check whether it shows the main menu if the right user name and password are entered.
Test Data	'Enter User Name' & 'Enter Password'
Expected Test Result	If the right user name and password are entered, main menu will be displayed.
Actual Test Result	As shown below
Conclusion	Successful. Main menu was shown when the right user name and password are entered.



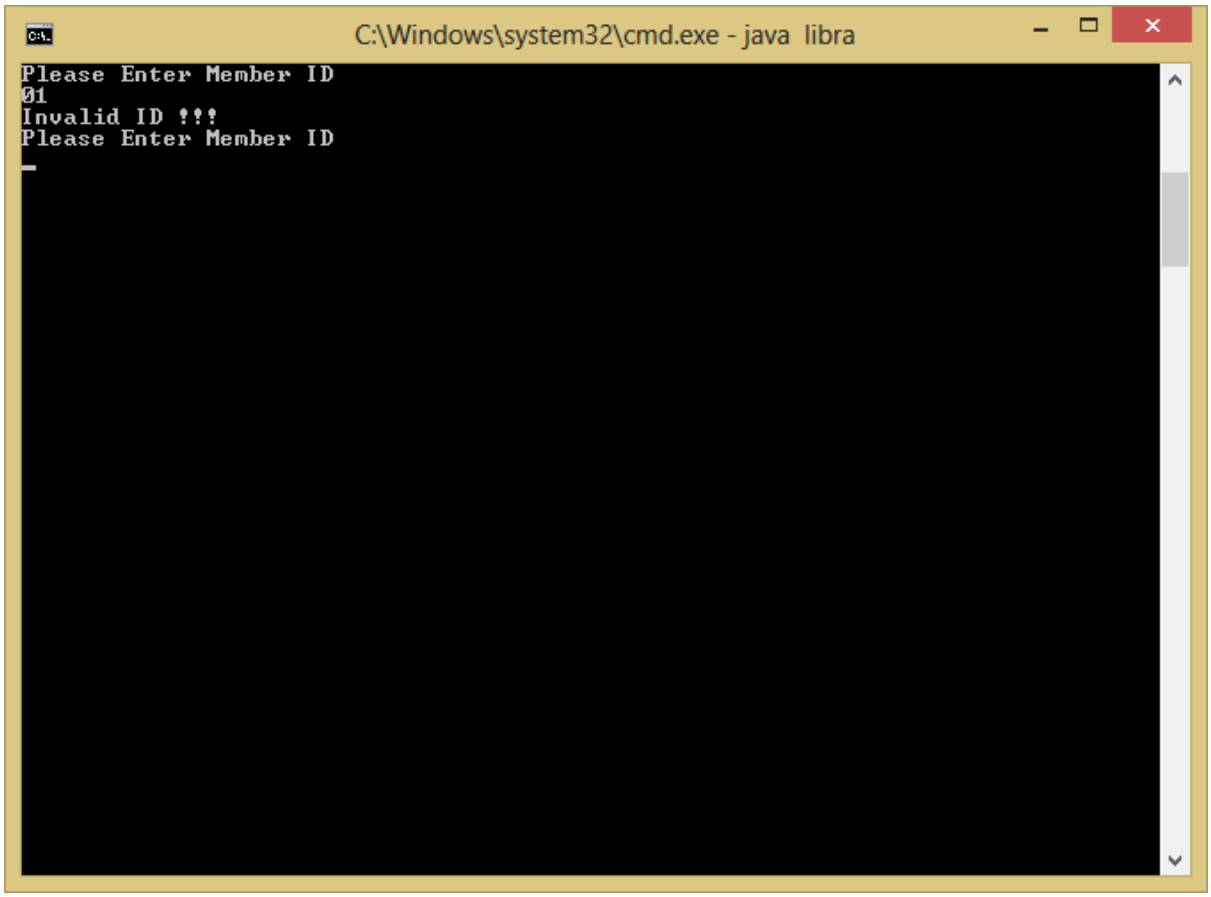
Test Case	4
Objectives	To check whether it shows the Add New Member function if the user enters 1 in main menu choice.
Test Data	Enter '1'
Expected Test Result	If the user enters 1, member ID will be prompted to enter.
Actual Test Result	As shown below
Conclusion	Successful. Please enter Member ID was shown.



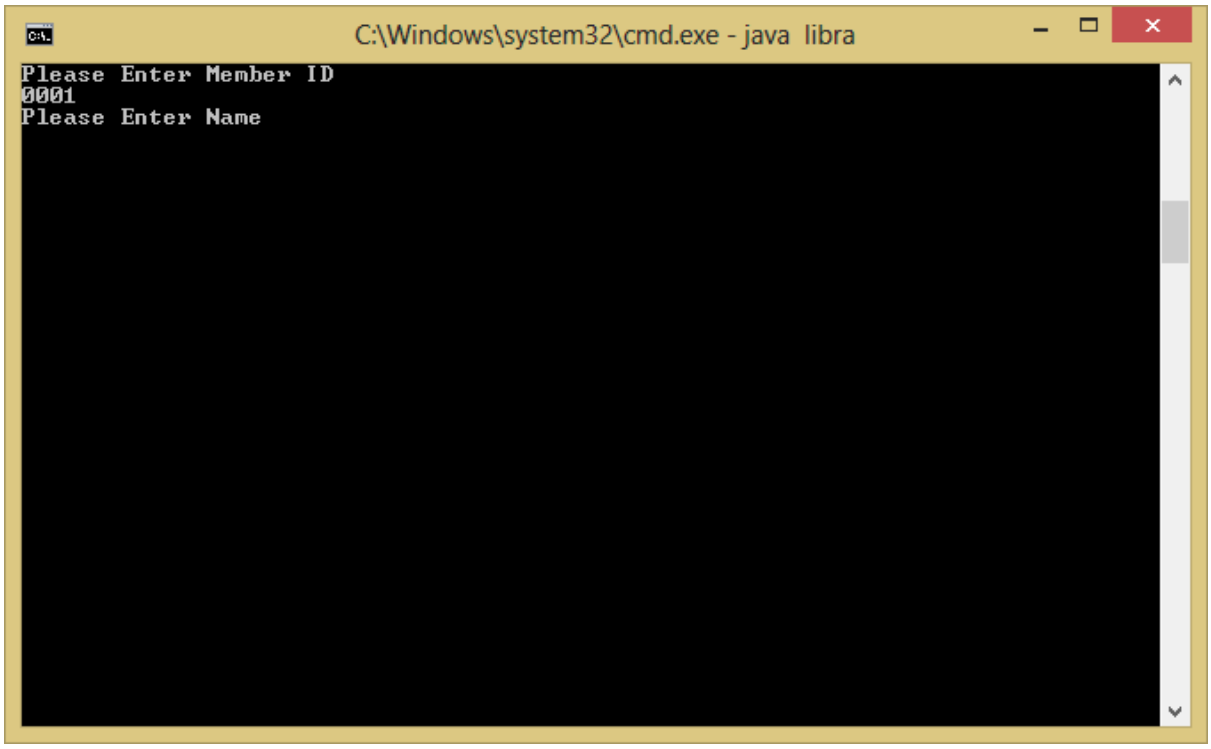
Test Case	5
Objectives	To check whether it shows error message if the user enters more than 4 characters in Member ID field.
Test Data	Enter '00001'
Expected Test Result	If the user enters 00001, error message will be displayed and member ID will be prompted to enter again.
Actual Test Result	As shown below
Conclusion	Successful. Error message was shown and 'Enter Member ID' was shown again.



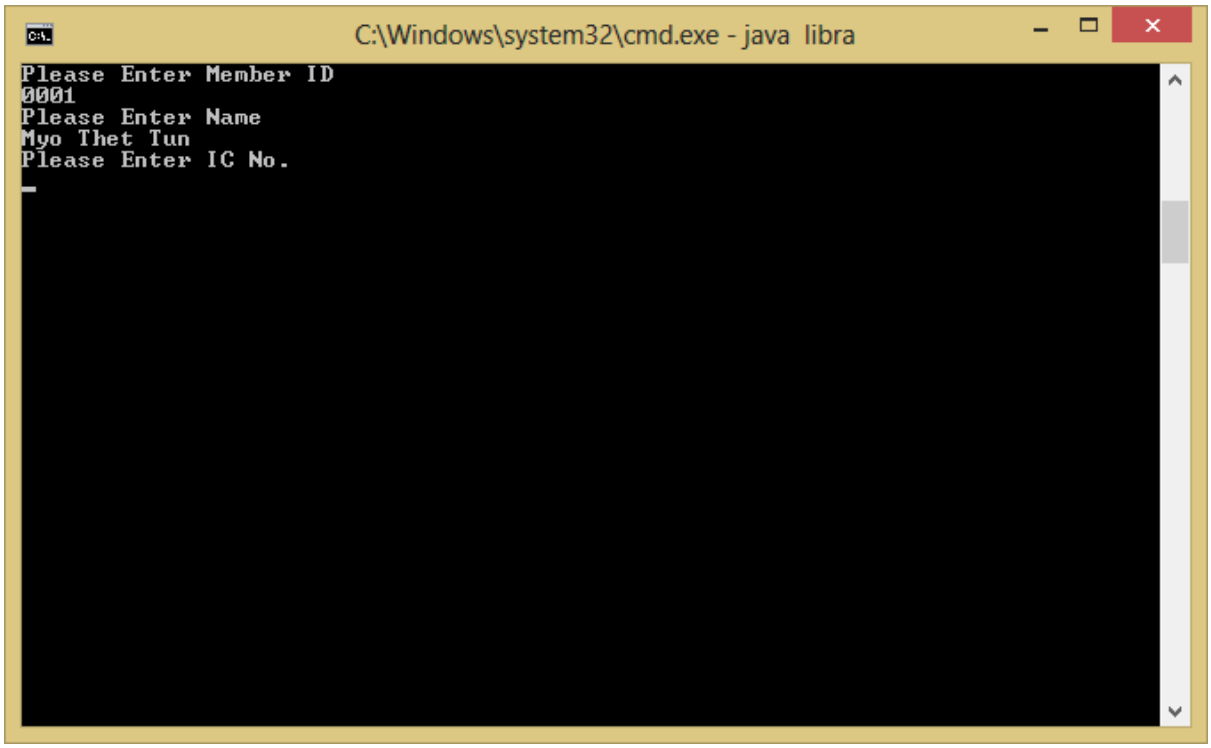
Test Case	6
Objectives	To check whether it shows error message if the user enters less than 4 characters in Member ID field.
Test Data	Enter '01'
Expected Test Result	If the user enters 01, error message will be displayed and member ID will be prompted to enter again.
Actual Test Result	As shown below
Conclusion	Successful. Error message was shown and 'Enter Member ID' was shown again.



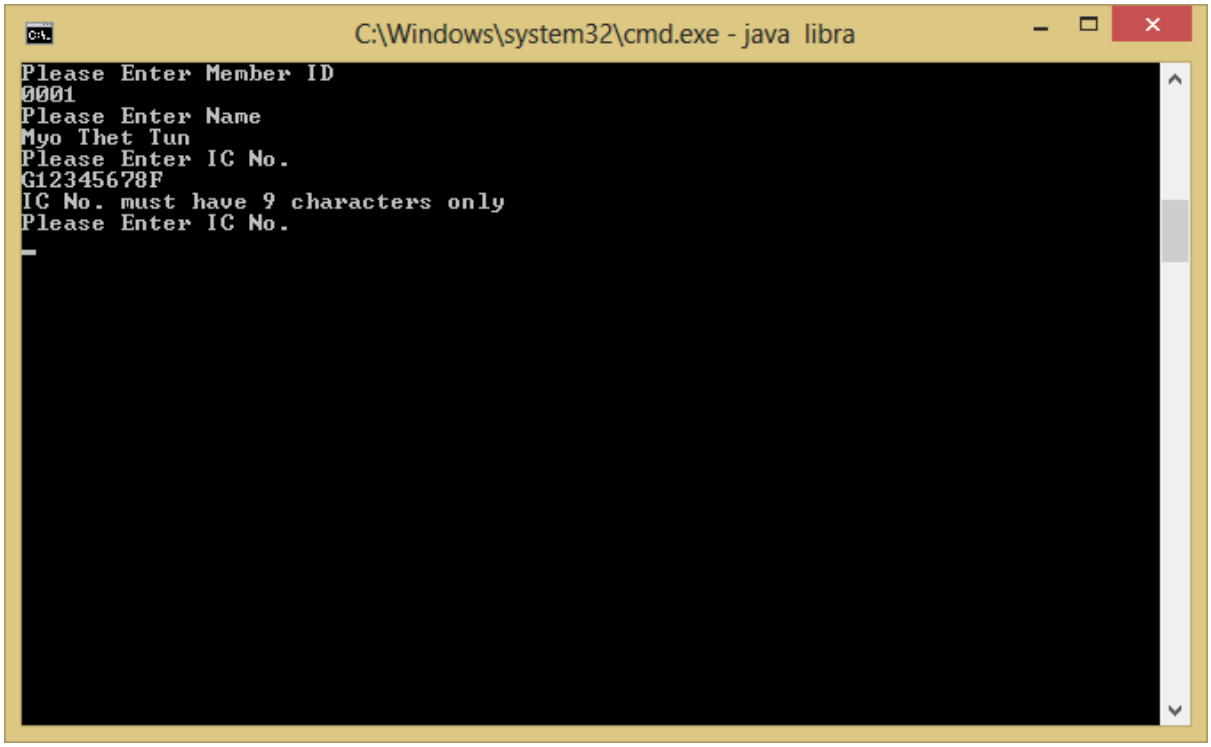
Test Case	7
Objectives	To check whether it prompt 'Enter name' if the user enters 4 characters in Member ID field.
Test Data	Enter '0001'
Expected Test Result	If the user enters 0001, 'Enter name' will be displayed.
Actual Test Result	As shown below
Conclusion	Successful. 'Enter name' was shown.



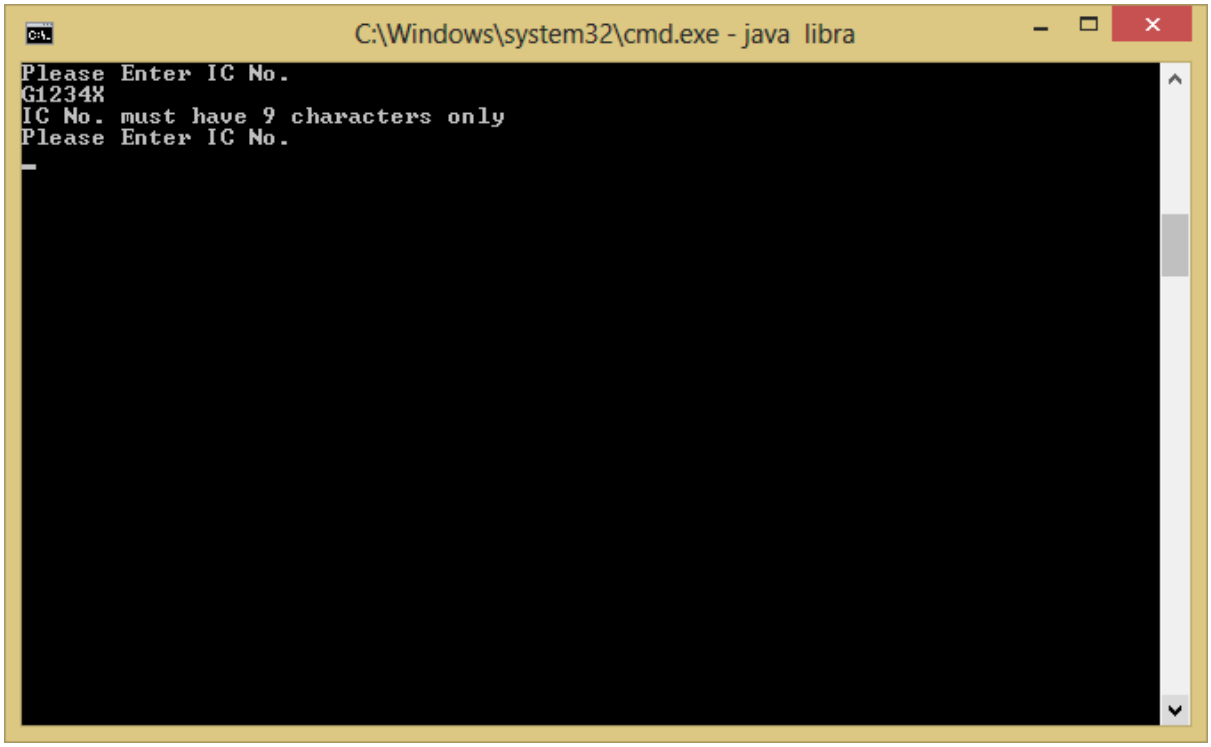
Test Case	8
Objectives	To check whether it prompt 'Enter IC No.' after the user enters name in Name field.
Test Data	Enter 'Myo Thet Tun'
Expected Test Result	If the user enters name, 'Enter IC No.' will be displayed.
Actual Test Result	As shown below
Conclusion	Successful. 'Enter IC No.' was shown.



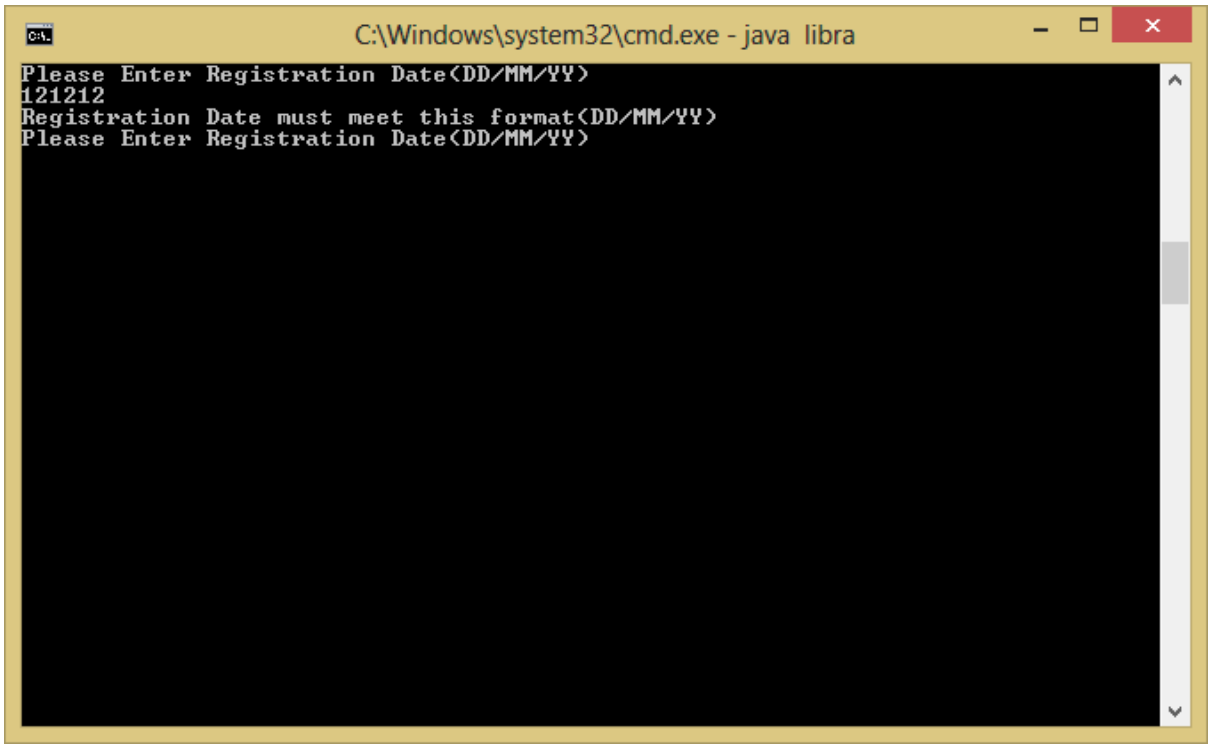
Test Case	9
Objectives	To check whether it shows error message if the user enters more than 9 characters in IC No. field.
Test Data	Enter 'G12345678F'
Expected Test Result	If the user enters 'G12345678F', error message will be displayed and IC No. will be prompted to enter again.
Actual Test Result	As shown below
Conclusion	Successful. Error message was shown and 'Enter IC No.' was shown again.



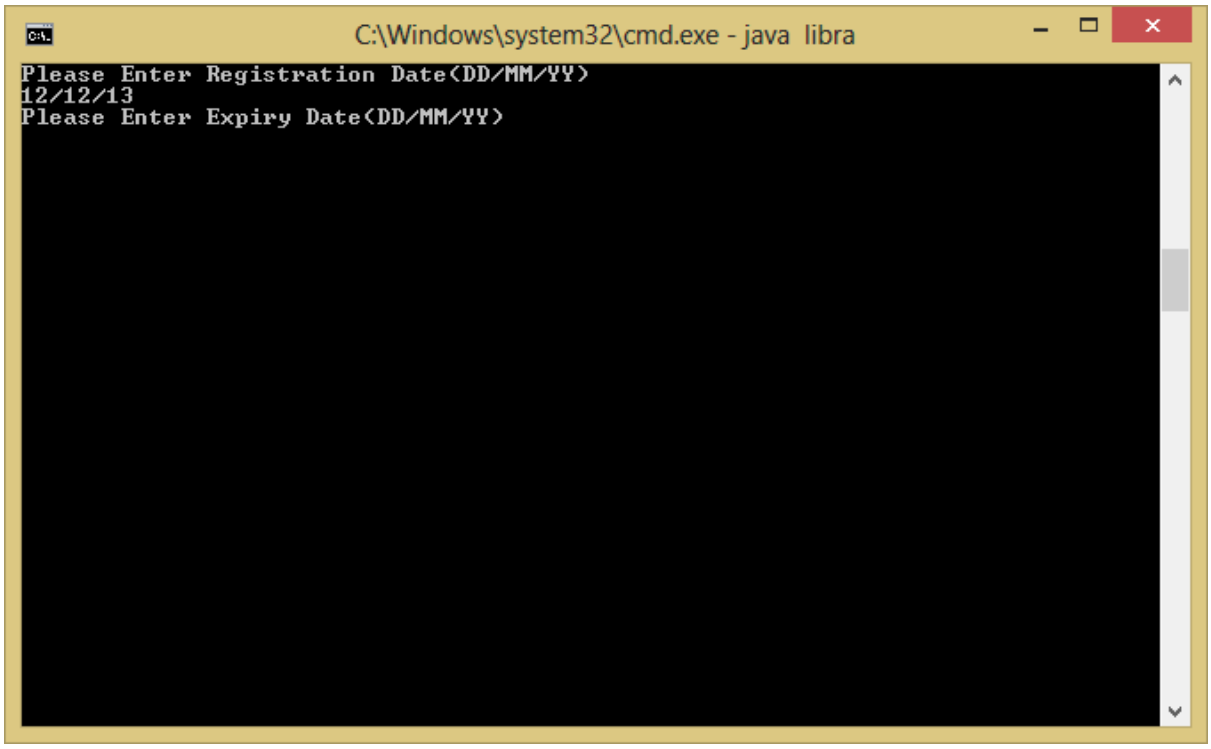
Test Case	10
Objectives	To check whether it shows error message if the user enters less than 9 characters in IC No. field.
Test Data	Enter 'G1234X'
Expected Test Result	If the user enters 'G1234X', error message will be displayed and IC No. will be prompted to enter again.
Actual Test Result	As shown below
Conclusion	Successful. Error message was shown and 'Enter IC No.' was shown again.



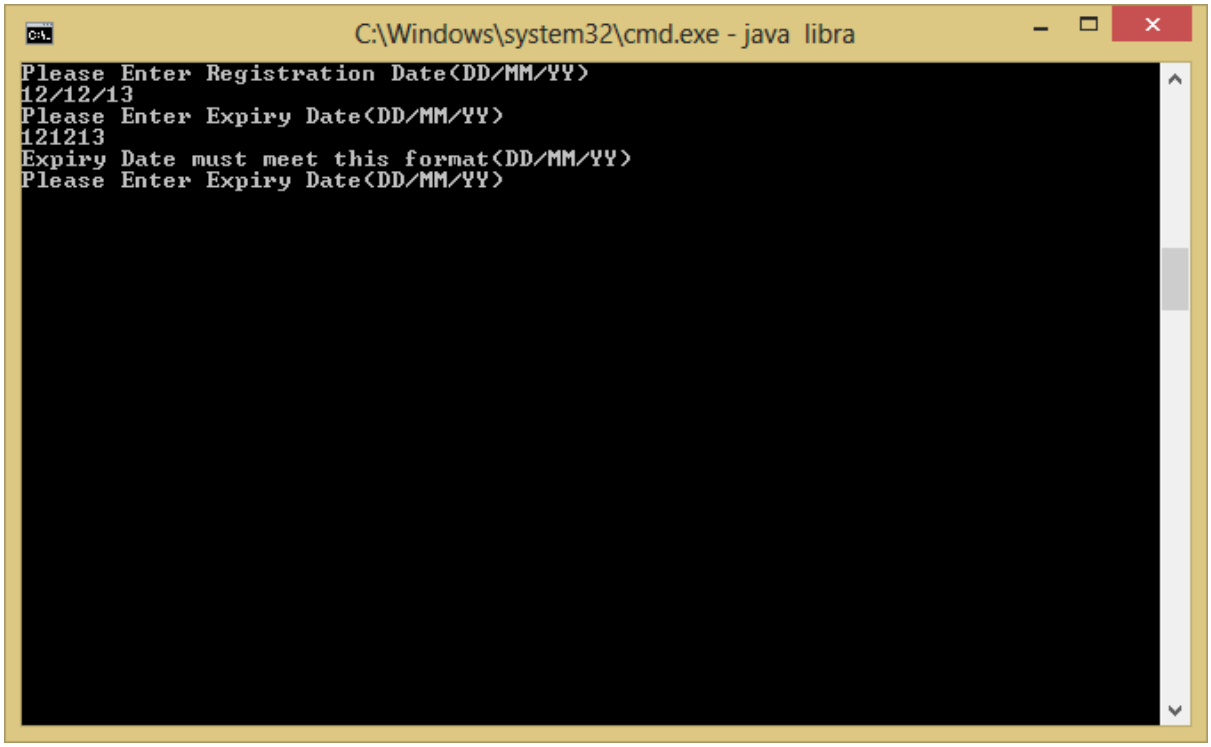
Test Case	11
Objectives	To check whether it shows error message if the user enters invalid date format in Registration Date field.
Test Data	Enter '121212'
Expected Test Result	If the user enters '121212', error message will be displayed and Registration Date will be prompted to enter again.
Actual Test Result	As shown below
Conclusion	Successful. Error message was shown and 'Enter Registration Date' was shown again.



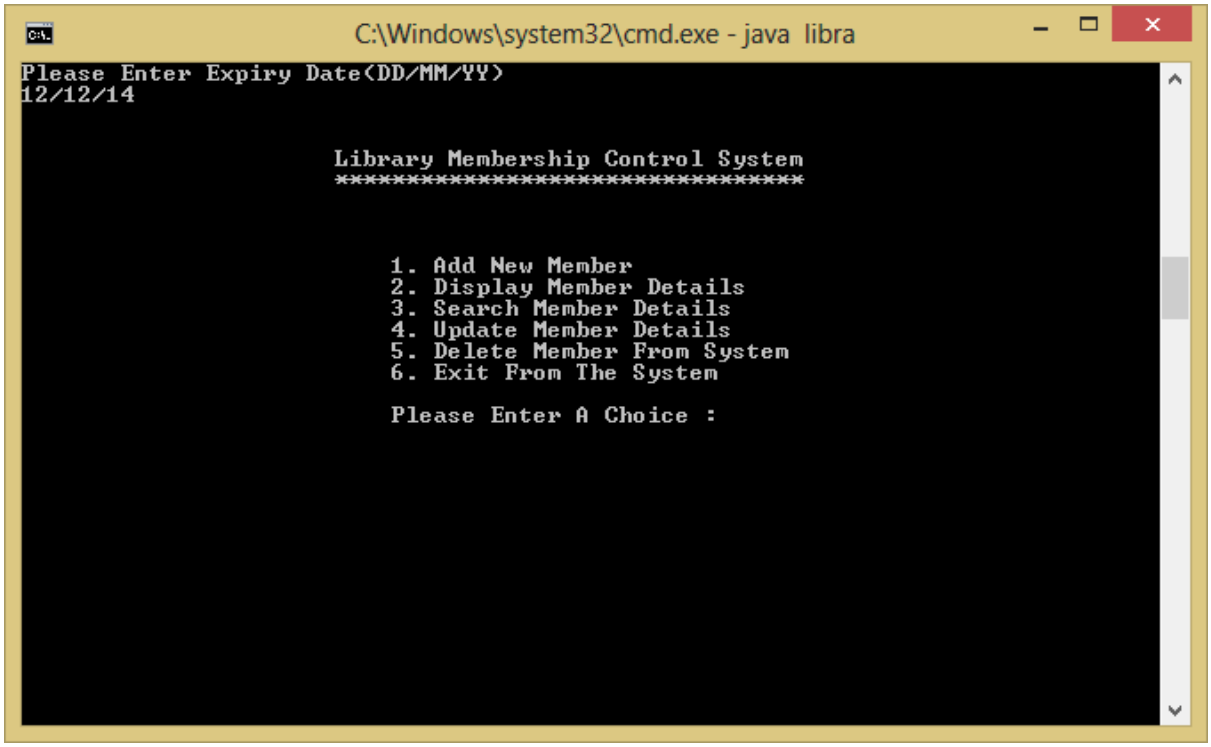
Test Case	12
Objectives	To check whether it prompt 'Enter Expiry Date' if the user enters valid date format in Registration Date field.
Test Data	Enter '12/12/13'
Expected Test Result	If the user enters '12/12/13', 'Enter expiry date' will be displayed.
Actual Test Result	As shown below
Conclusion	Successful. 'Enter expiry date' was shown.



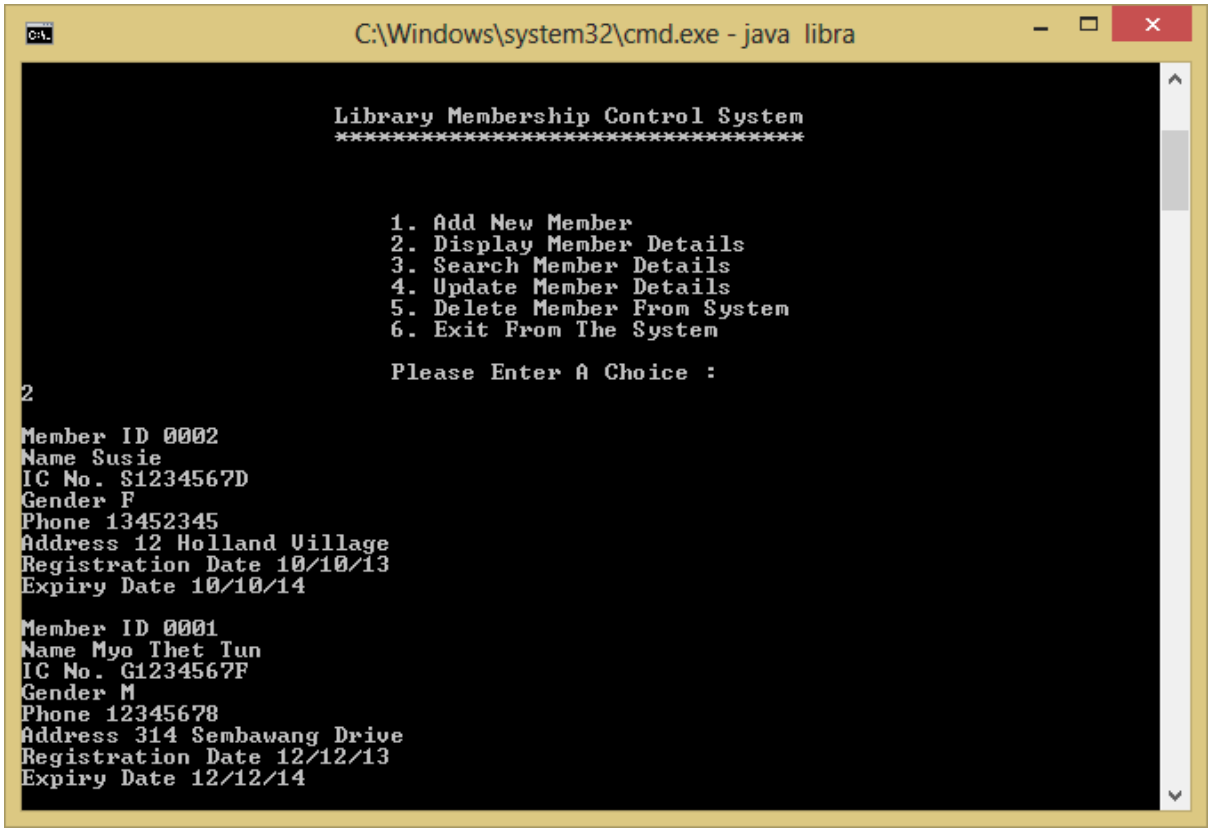
Test Case	13
Objectives	To check whether it shows error message if the user enters invalid date format in Expiry Date field.
Test Data	Enter '121213'
Expected Test Result	If the user enters '121213', error message will be displayed and expiry date will be prompted to enter again.
Actual Test Result	As shown below
Conclusion	Successful. Error message was shown and 'Enter Expiry Date' was shown again.



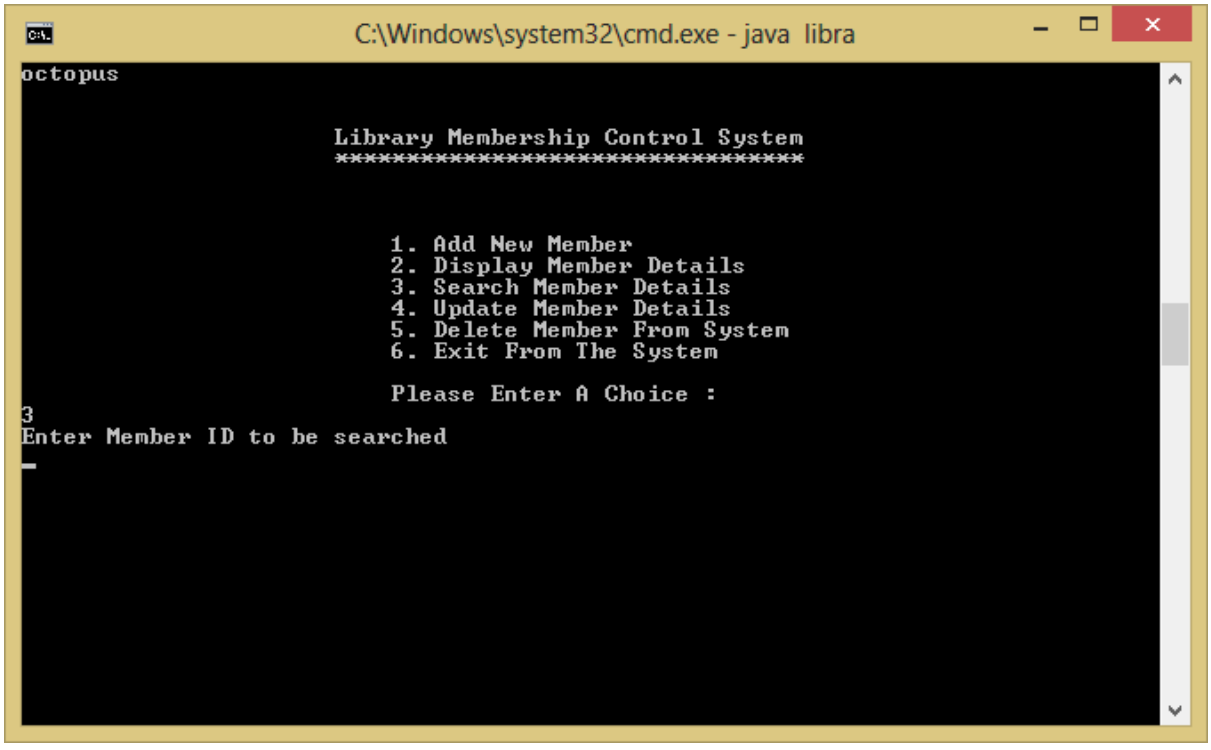
Test Case	14
Objectives	To check whether it shows Main Menu if the user enters valid date format in Expiry Date field.
Test Data	Enter '12/12/14'
Expected Test Result	If the user enters '12/12/14', 'Main Menu' will be displayed.
Actual Test Result	As shown below
Conclusion	Successful. 'Main Menu' was shown.



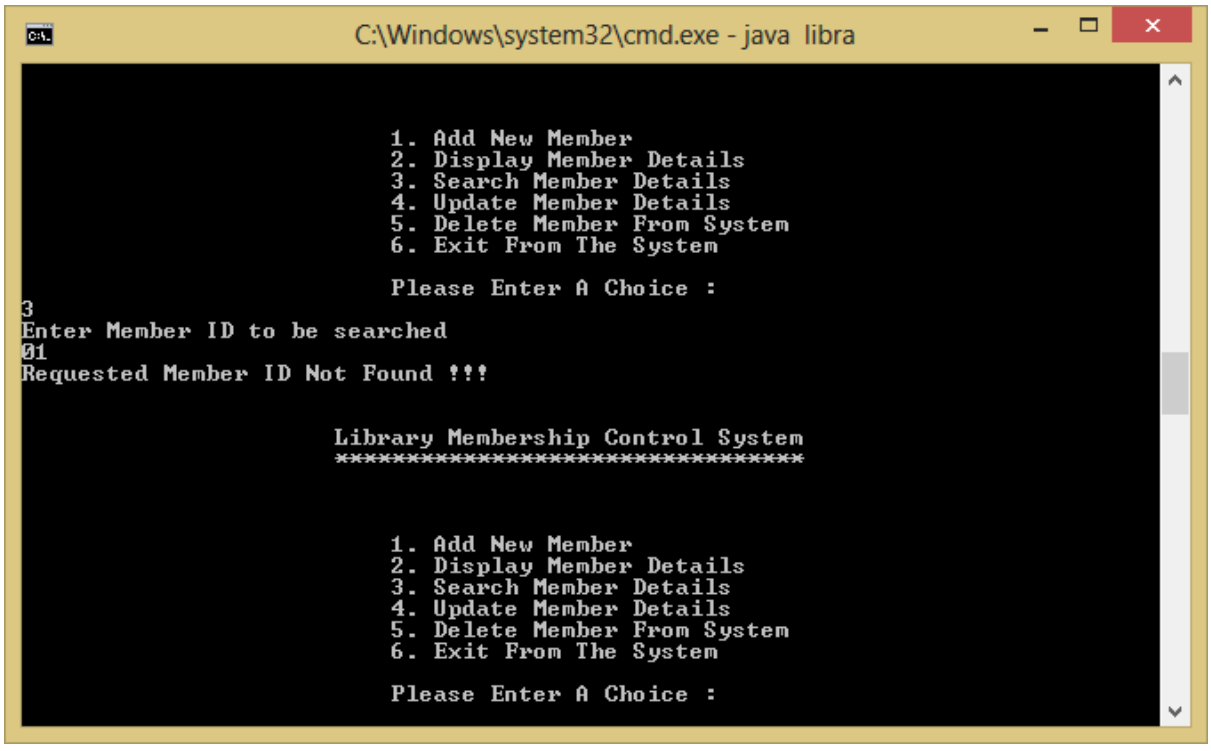
Test Case	15
Objectives	To check whether it shows the Display Member Details function if the user enters 2 in main menu choice.
Test Data	Enter '2'
Expected Test Result	If the user enters 2, all available member details will be displayed.
Actual Test Result	As shown below
Conclusion	Successful. All available member details were shown.



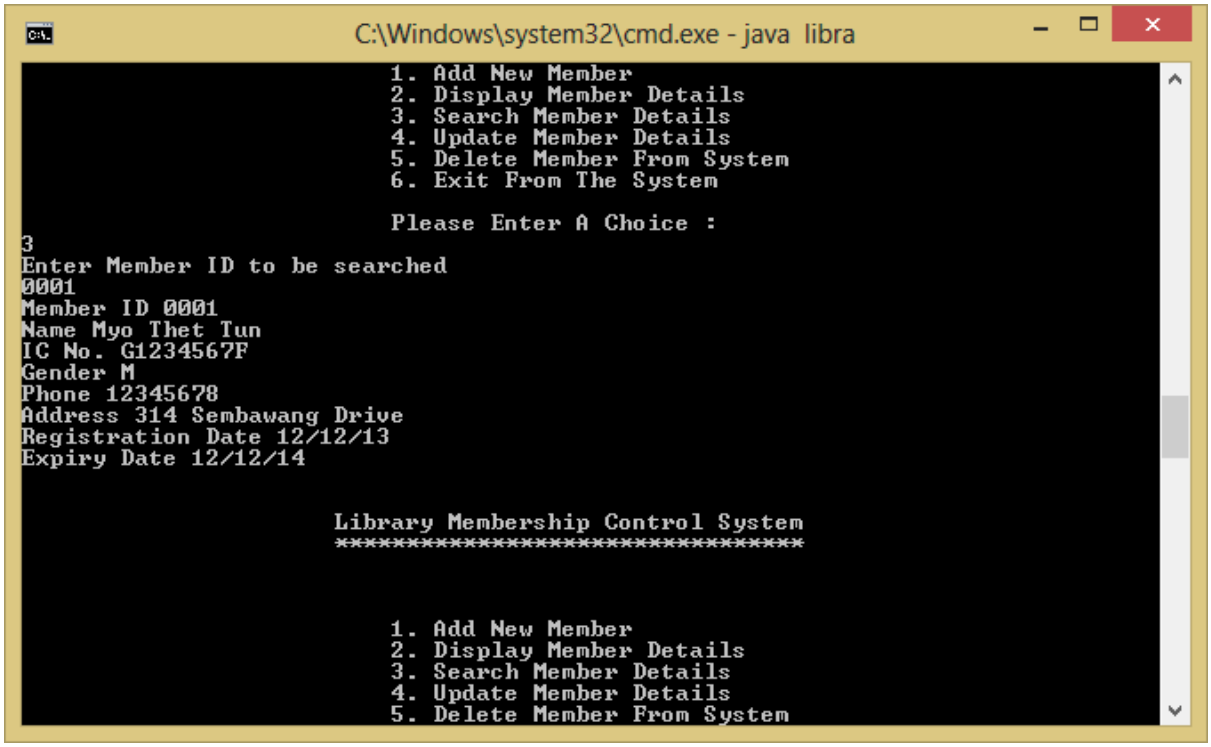
Test Case	16
Objectives	To check whether it shows the Search Member Details function if the user enters 3 in main menu choice.
Test Data	Enter '3'
Expected Test Result	If the user enters 3, member ID will be requested to enter
Actual Test Result	As shown below
Conclusion	Successful. 'Enter member ID to be searched' was prompted.



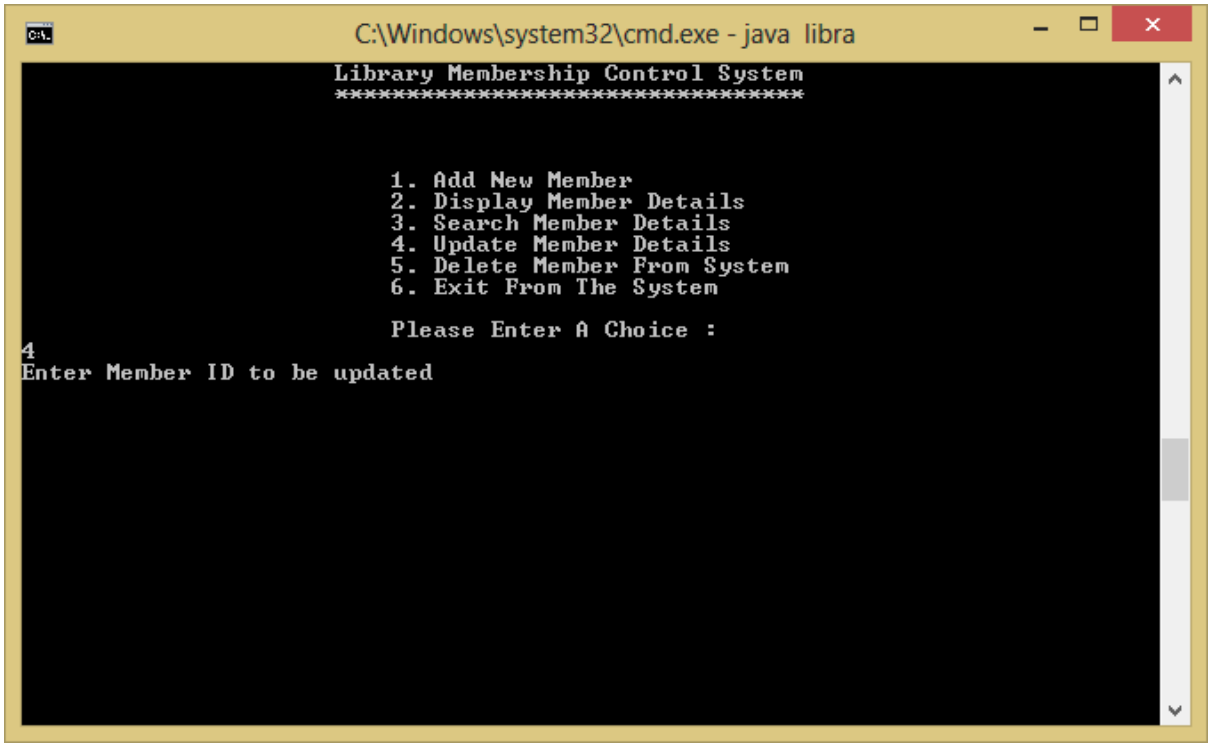
Test Case	17
Objectives	To check whether it shows error message if the user enters unavailable ID in search member function.
Test Data	Enter '01'
Expected Test Result	If the user enters 01, error message will be shown and will go back to main menu.
Actual Test Result	As shown below
Conclusion	Successful. Error message was shown and main menu was displayed again.



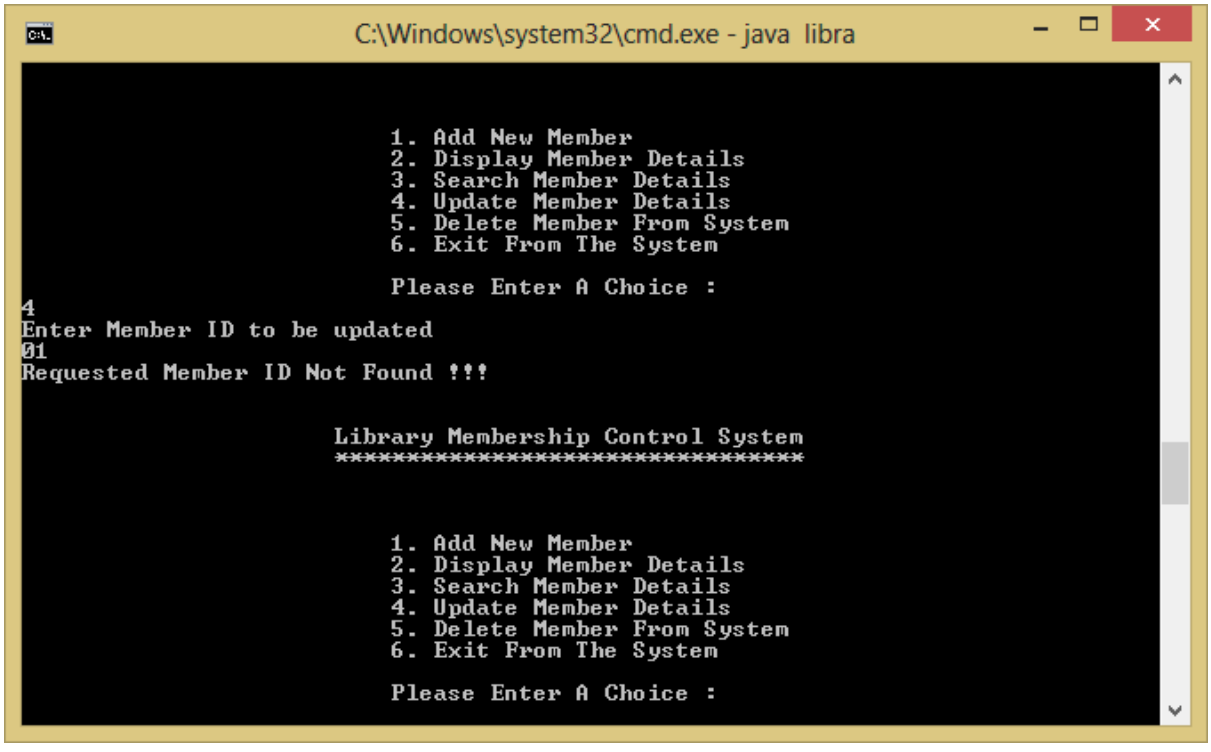
Test Case	18
Objectives	To check whether it shows member details if the user enters available ID in search member function.
Test Data	Enter '0001'
Expected Test Result	If the user enters 0001, member details will be shown and will go back to main menu.
Actual Test Result	As shown below
Conclusion	Successful. The details of member ID 0001 was shown and main menu was displayed again.



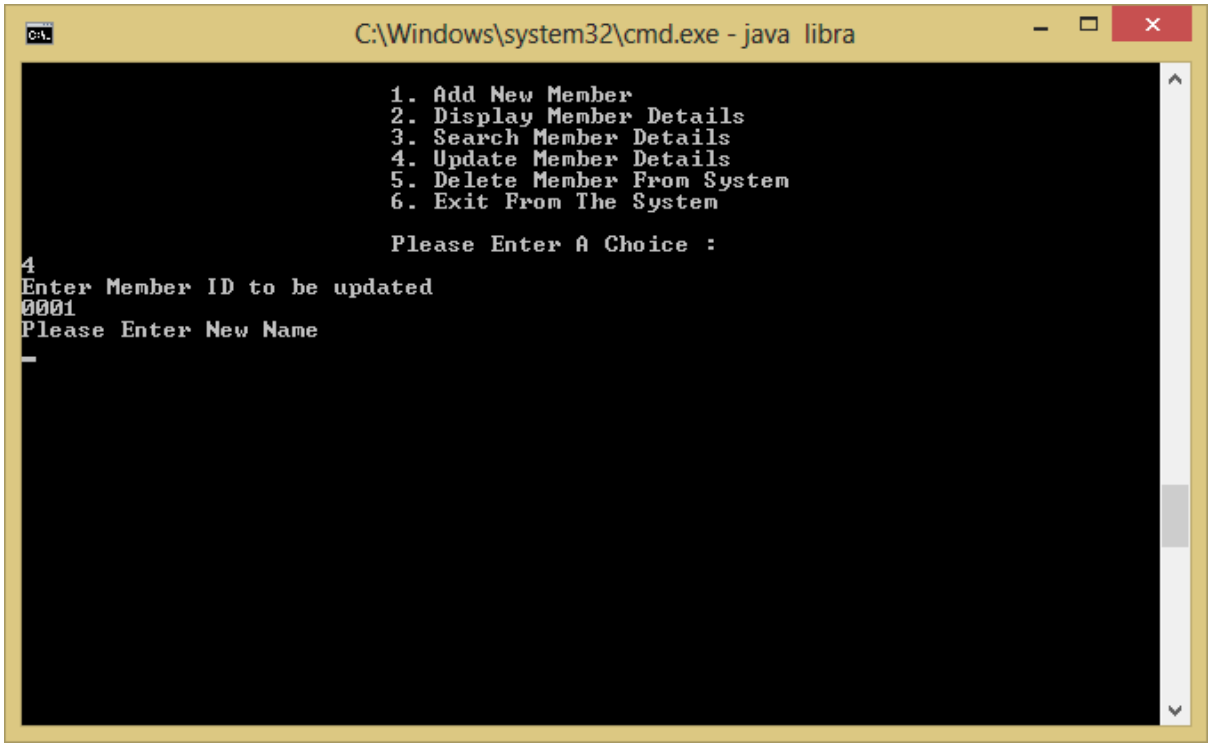
Test Case	19
Objectives	To check whether it shows the Update Member Details function if the user enters 4 in main menu choice.
Test Data	Enter '4'
Expected Test Result	If the user enters 4, member ID will be requested to enter
Actual Test Result	As shown below
Conclusion	Successful. 'Enter member ID to be updated' was prompted.



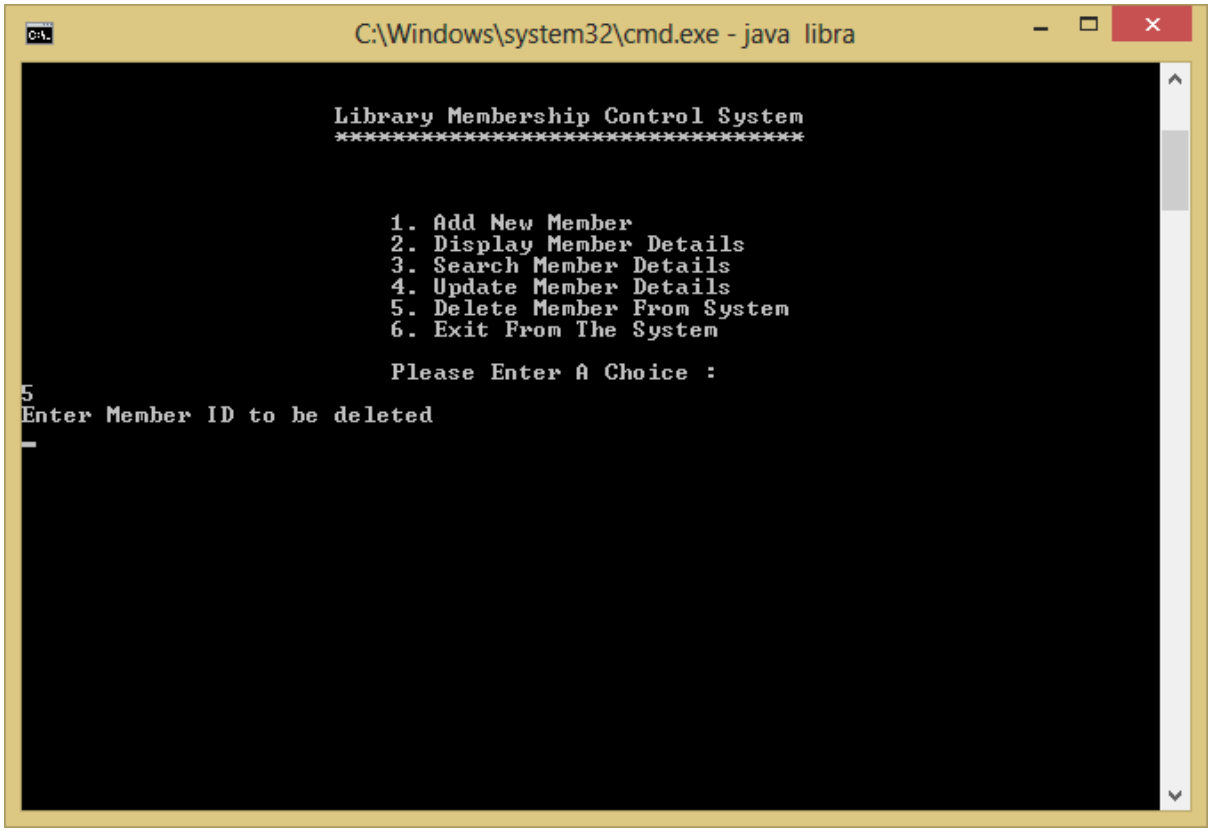
Test Case	20
Objectives	To check whether it shows error message if the user enters unavailable ID in update member function.
Test Data	Enter '01'
Expected Test Result	If the user enters 01, error message will be shown and will go back to main menu.
Actual Test Result	As shown below
Conclusion	Successful. Error message was shown and main menu was displayed again.



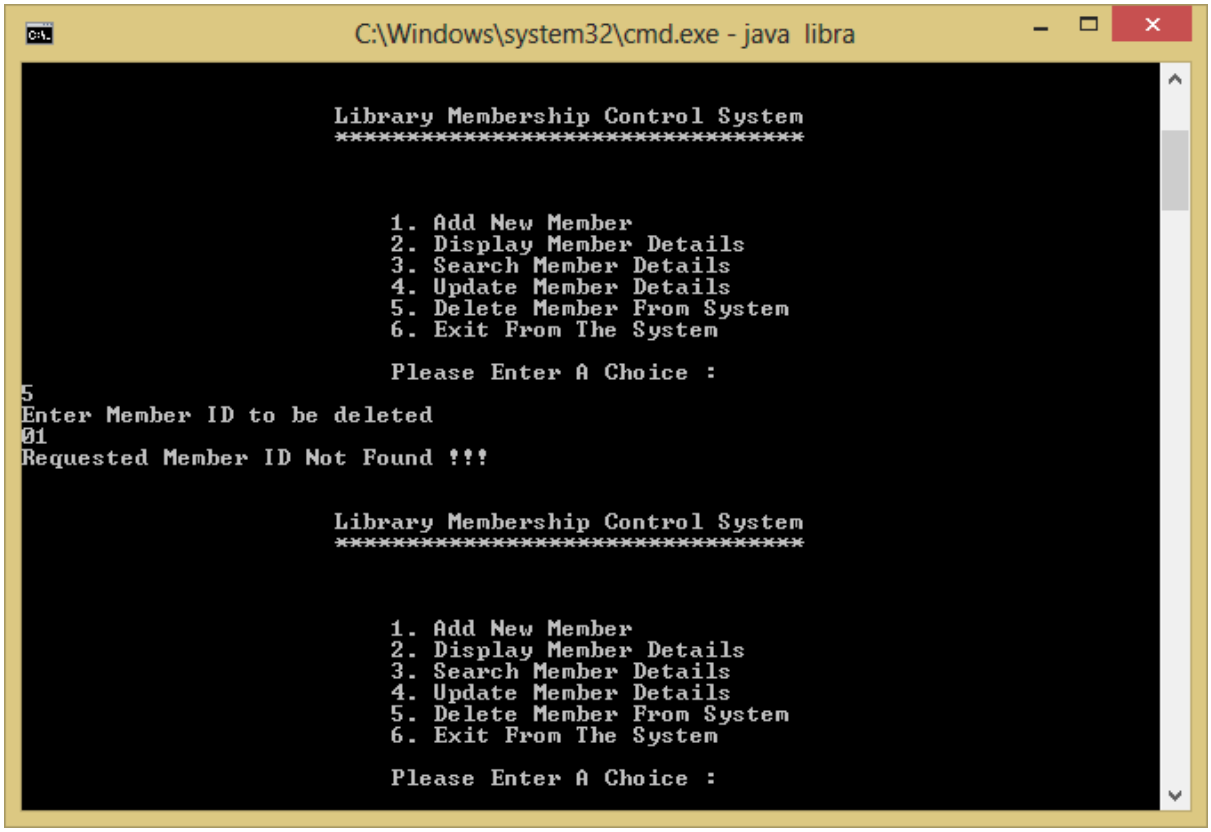
Test Case	21
Objectives	To check whether it prompts to update member details if the user enters available ID in update member function.
Test Data	Enter '0001'
Expected Test Result	If the user enters 0001, 'Enter new name' will be shown.
Actual Test Result	As shown below
Conclusion	Successful. It prompted to enter new name for Member ID 0001.



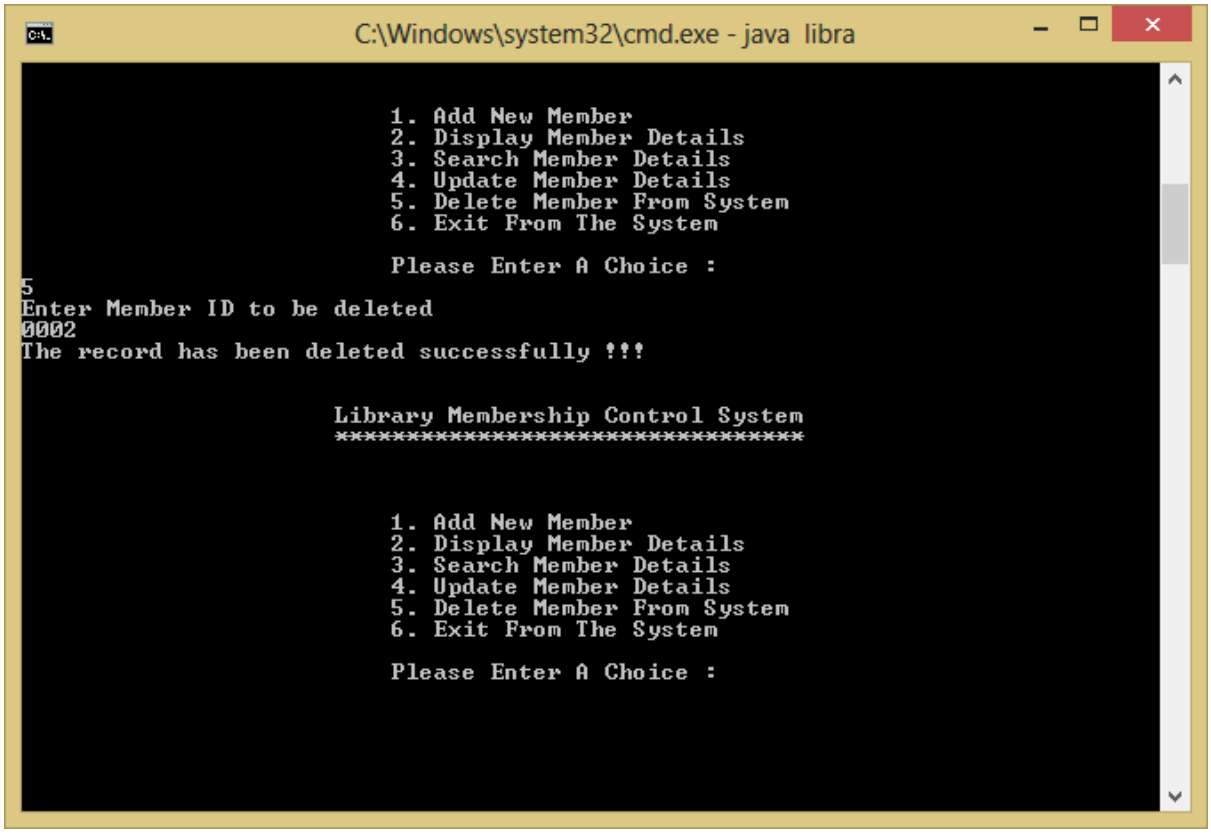
Test Case	22
Objectives	To check whether it shows the Delete Member function if the user enters 5 in main menu choice.
Test Data	Enter '5'
Expected Test Result	If the user enters 5, member ID will be requested to enter.
Actual Test Result	As shown below
Conclusion	Successful. 'Enter member ID to be deleted' was prompted.



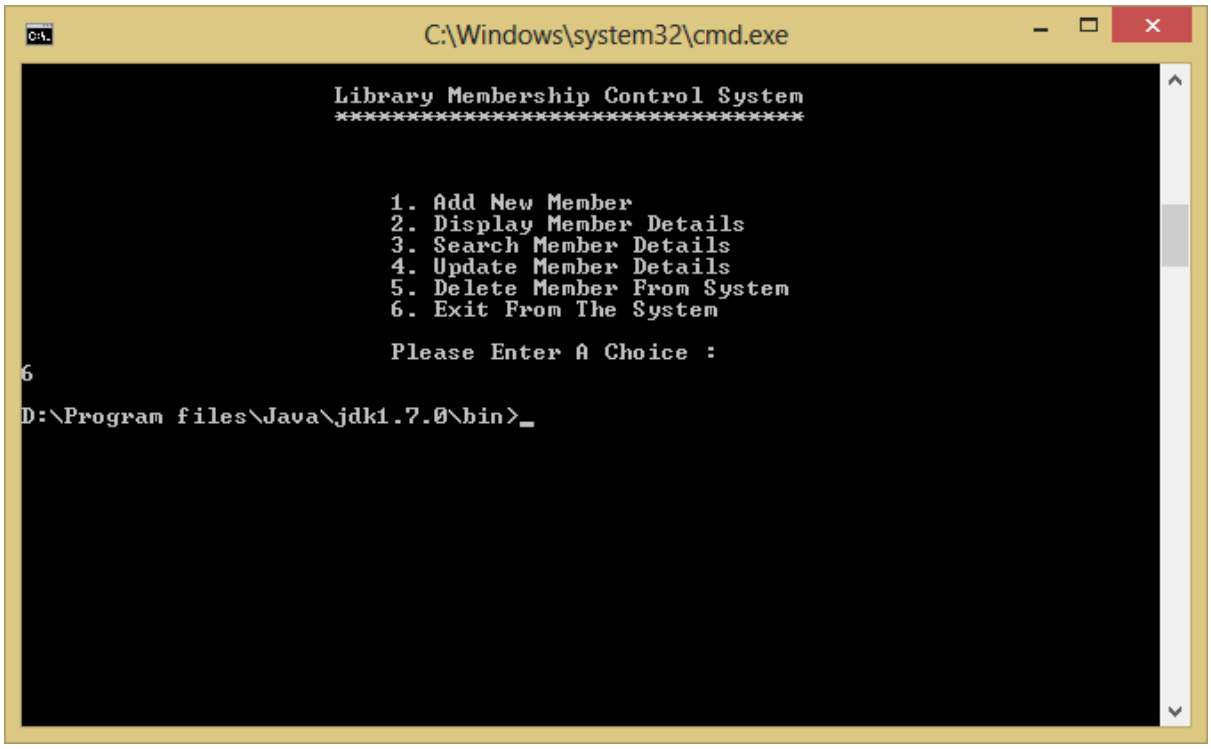
Test Case	23
Objectives	To check whether it shows error message if the user enters unavailable ID in delete member function.
Test Data	Enter '01'
Expected Test Result	If the user enters 01, error message will be shown and will go back to main menu.
Actual Test Result	As shown below
Conclusion	Successful. Error message was shown and main menu was displayed again.



Test Case	24
Objectives	To check whether it displays the record is deleted if the user enters available ID in delete member function.
Test Data	Enter '0002'
Expected Test Result	If the user enters 0002, 'The record has been deleted successfully' will be shown and will go back to main menu.
Actual Test Result	As shown below
Conclusion	Successful. 'The record has been deleted successfully' was shown and main menu was displayed again.



Test Case	25
Objectives	To check whether it exits from the system if the user enters 6 in main menu choice.
Test Data	Enter '6'
Expected Test Result	If the user enters 6, the system will be closed.
Actual Test Result	As shown below
Conclusion	Successful. The system was exited.



Test Log

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

4. IMPLEMENTATION

4.1 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)

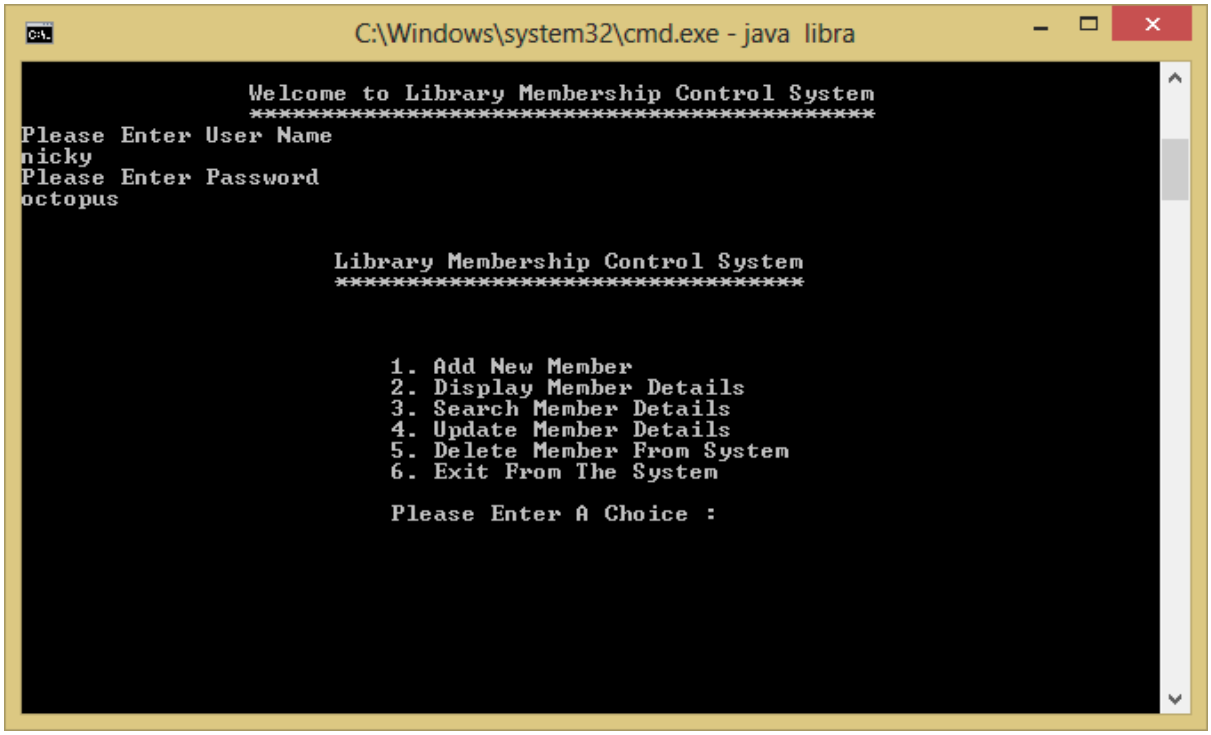
Software requirements:

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

Java Development Kit (JDK) version 6 (jdk 1.6) or 7 (jdk 1.7)

CD-ROM Drive

4.2 User Manual



'Welcome Screen' appears.

Enter user name 'nicky' and 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.

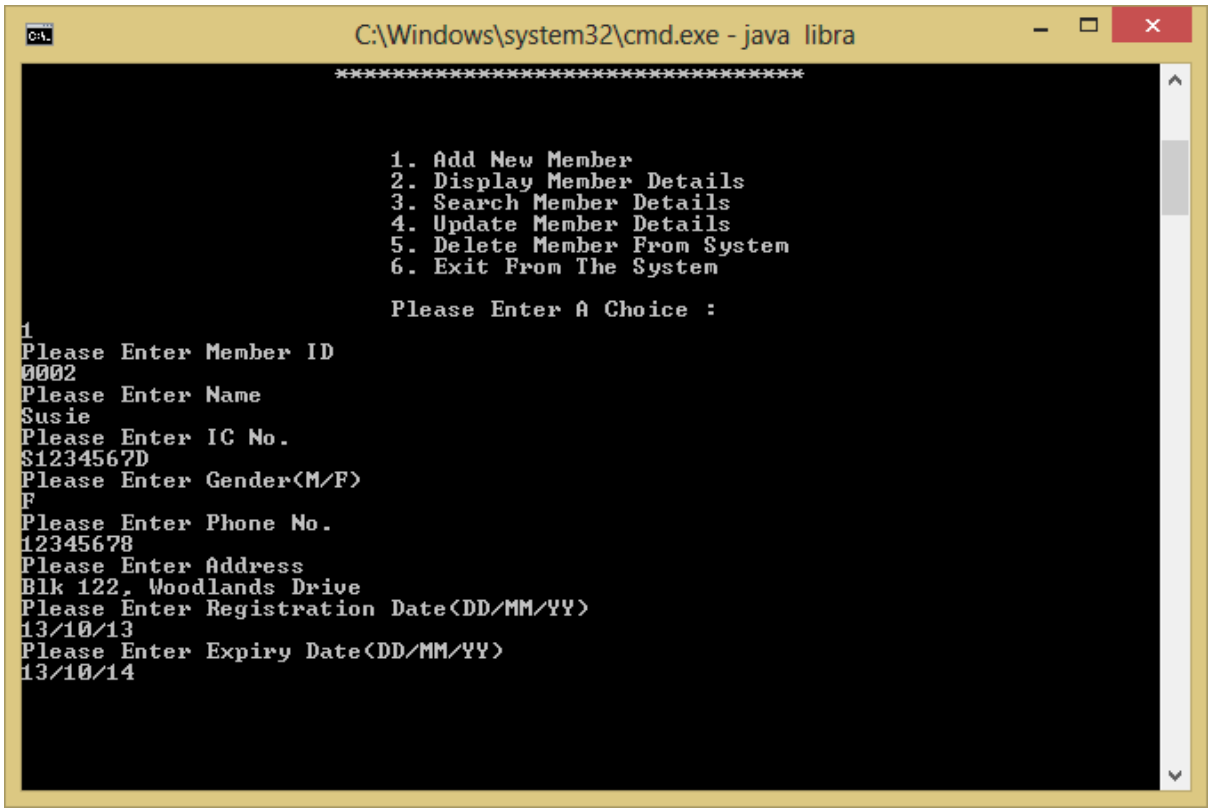
Type 3 to search for a specific member using Member ID.

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 must be less than 40 characters.

IC No. must have 9 characters.

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

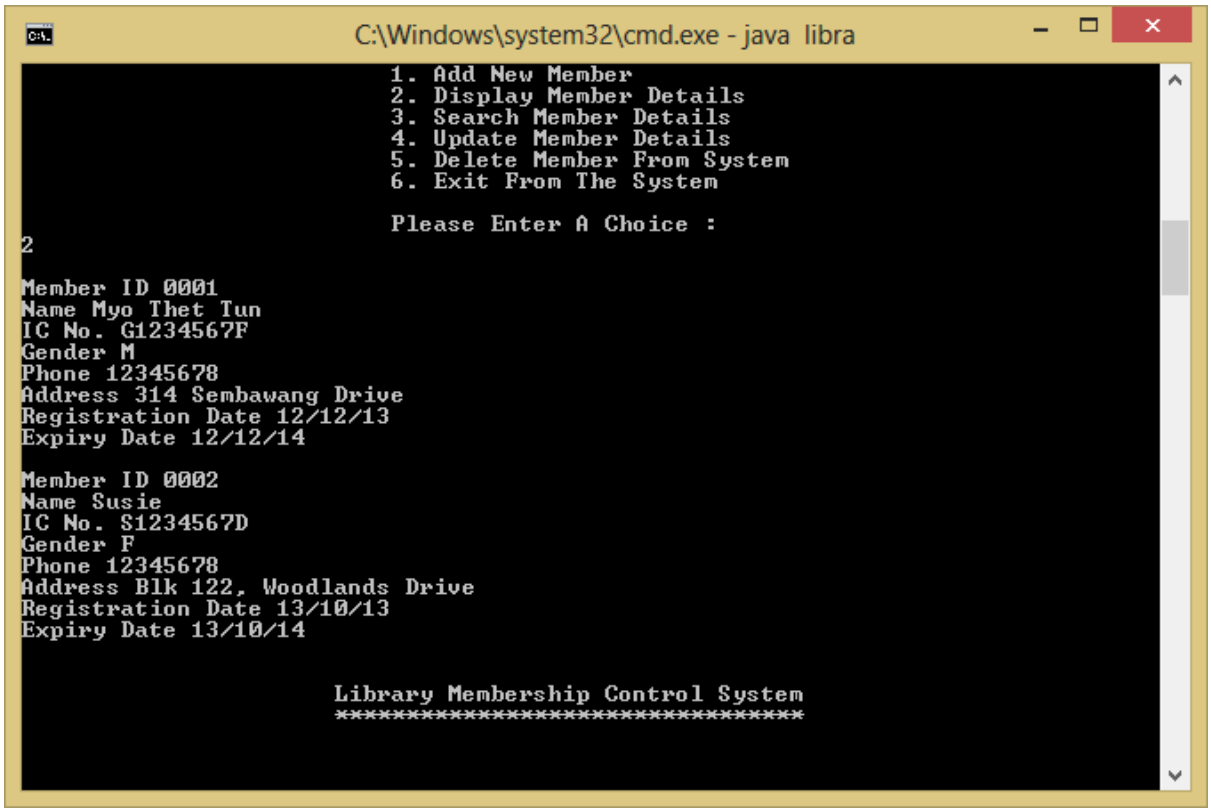
Phone No. must have 8 characters.

Address must be less than 60 characters.

Registration Date must have 8 characters and must follow this date format (DD/MM/YY).

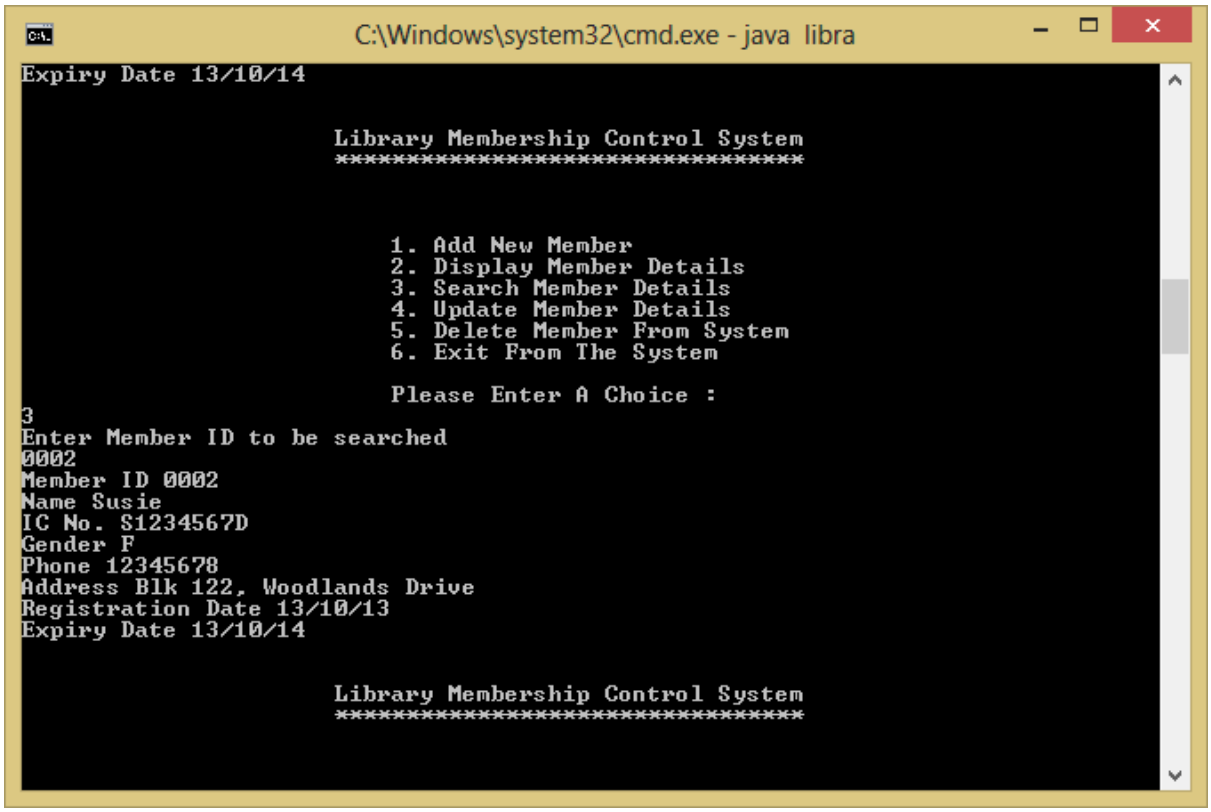
Expiry Date must have 8 characters and must follow this date format (DD/MM/YY).

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



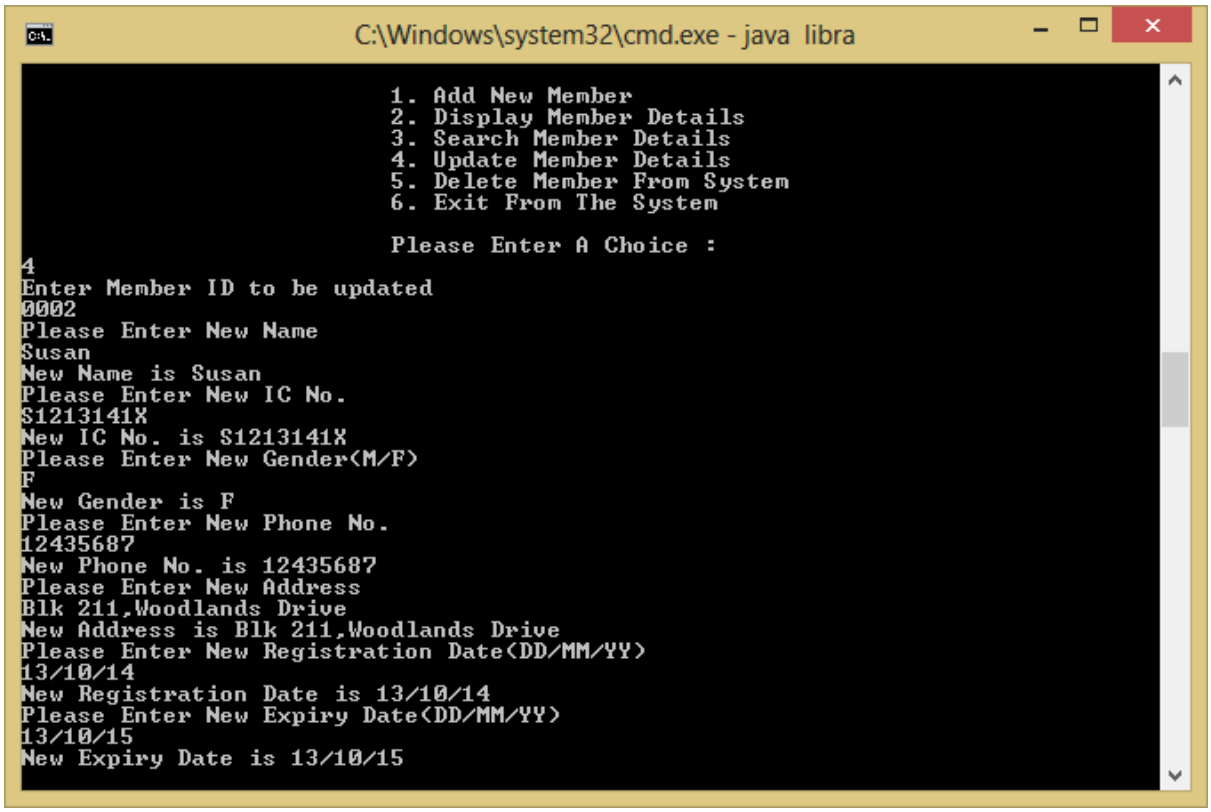
Currently, there are 2 members, Member ID 0001 and 0002, in the file. And their details will be displayed.

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. 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.

Name must be less than 40 characters.

IC No. must have 9 characters.

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

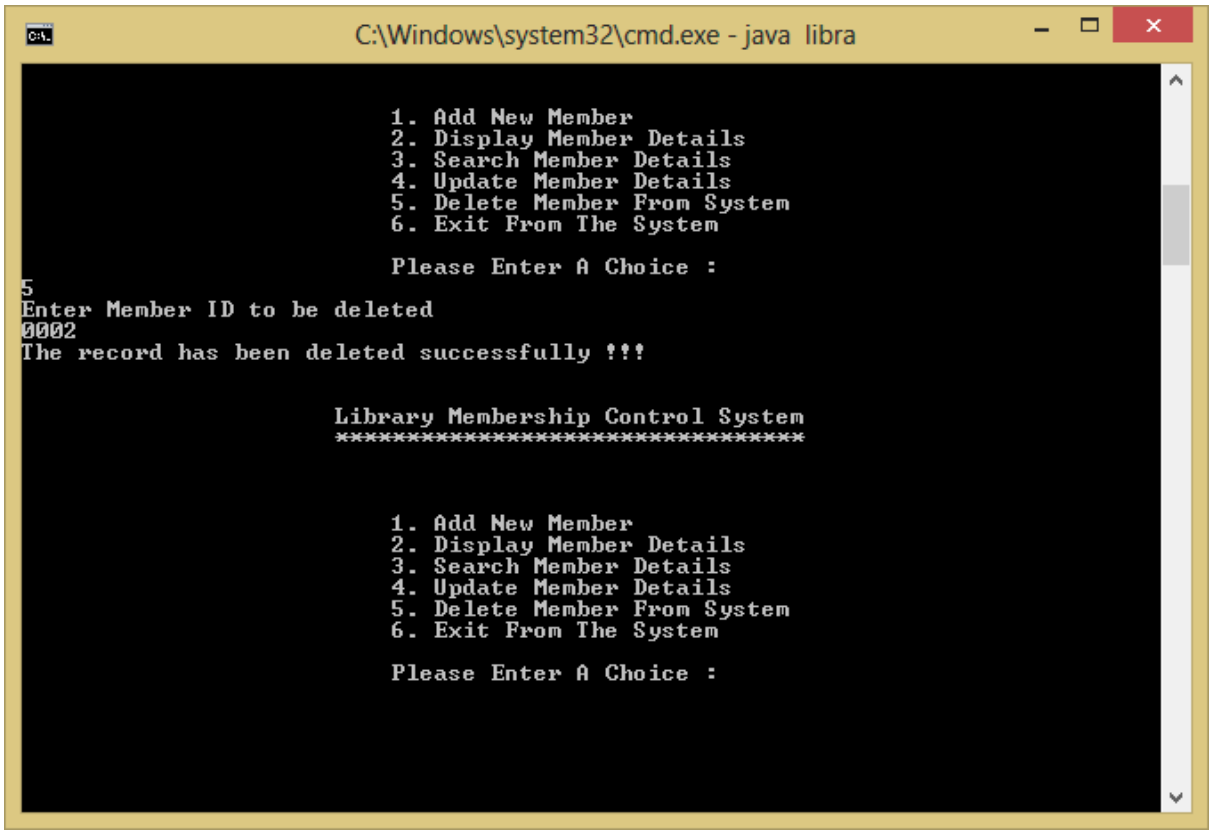
Phone No. must have 8 characters.

Address must be less than 60 characters.

Registration Date must have 8 characters and must follow this date format (DD/MM/YY).

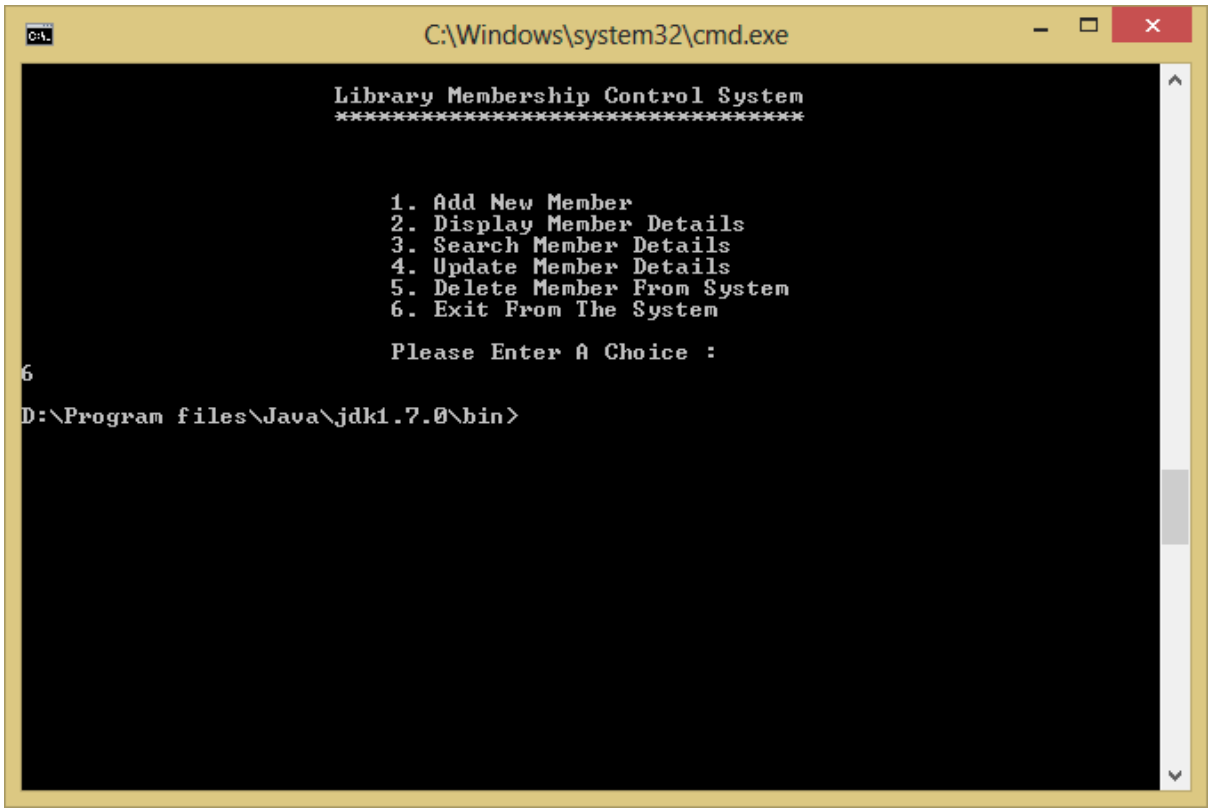
Expiry Date must have 8 characters and must follow this date format (DD/MM/YY).

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.

5. CONCLUSION

5.1 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.

5.2 Program Weaknesses

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

5.3 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.