



**UTHM**  
Universiti Tun Hussein Onn Malaysia

**FACULTY OF COMPUTER SCIENCE AND INFORMATION  
TECHNOLOGY**

**BIT 10703  
DATA STRUCTURE AND ALGORITHM**

*LECTURER NAME  
PN.ROZLINI BINTI MOHAMED*

*SECTION: 3*

**PROJECT TITLE:**  
**CLUB MEMBER RESGISTRATION SYSTEM FOR IT CLUB  
FSKTM**

NAME	MATRIC NUMBER
AHMAD SYAWAL BIN AHMAD ZAIDI	AI210282
CHIA SZE YEE	AI210360
KOR WEI YAN	AI210411
MUHAMMAD ALI UZAIR BIN MARZELE	AI210178
NURAFIQAH SHERLY BINTI ZAINI	AI210300
NUR AISYAH ZAYANI BINTI HAMID	AI210063

## CONTENTS

<b>1.0 INTRODUCTION .....</b>	<b>1</b>
<b>2.0 OBJECTIVES .....</b>	<b>1</b>
<b>3.0 PROBLEM BACKGROUND.....</b>	<b>2</b>
<b>4.0 METHODOLOGY.....</b>	<b>2-13</b>
<b>5.0 RESULT.....</b>	<b>13-21</b>
<b>6.0 CONCLUSION.....</b>	<b>21</b>

## **1.0 INTRODUCTION**

Under the Faculty of Computer Science and Information Technology (FSKTM), the Information Technology Club (ITC) was established. ITC was established to gather the students in FSKTM from different courses to organise activities and create a greater bonding between the students of the same faculty. There are a total of 5 different courses under FSKTM, which includes Information Technology (BIT), Computer Science in Software Engineering (BIP), Computer Science in Multimedia (BIM), Computer Science in Web Technology (BIW), and Computer Science in Information Security (BIC).

However, due to rapid changes of club member's study year and semester; and also, the increment of the number of members in ITC, it has caused a problem for the club to trace all the registered members under the club. Therefore, ITC has engaged and invested in the development of Club Registration System for IT Club FSKTM.

Therefore, Club Member Registration System for IT Club FSKTM is a very systematic and convenient program because it allows the person in charge of the club to manage the members' details efficiently and systematically. It can also track the total registrations done and the current active members through the analysis after every registration is done successfully. It will also speed up the registration process and shorten the time consumed through manual registration.

## **2.0 OBJECTIVES**

There are few objectives upon the development of the Information Technology Club Registration System for FSKTM students. The objectives are as follows:

- (a) To record the details of the members systematically and efficiently.
- (b) To analyse the number of total registrations done.
- (c) To obtain the number of current active club members.

This approach benefits not only person in charge but also regular members, allowing ordinary members to connect and communicate with members of the post. The system might also bring advantages in terms of working efficiency and organisation. Therefore, this technique eliminates the need for data to be filed on paper and instead be recorded utilising a digital system.



### **3.0 PROBLEM BACKGROUND**

Currently, IT Club FSKTM still uses manual system to record the members student information. The files and documents were difficult to have a constant followed up in order to keep them up-to-date. This implies that if they overlook any information by accident, they will be labelled as having a poor management registration system, and it is impossible for the person in charge to locate the student information if it is assembled in a large number of files. Besides, the members of the club are constantly changing every semester and every study year. Without a proper data management system, the club will never get to proper file the changes of each member's data, and the details of the new members every semester.

Hence, a Club Registration System for IT Club FSKTM has been invested and developed to overcome these problems. First of all, in the Club Registration System, the functions included are insert a new record, remove a record, update a record, find record, and display record. To insert record, the student's information will be needed to complete the record insertion. Next, to remove or update a record, the record is first searched through matric number or I/C number before proceeding to remove or update. The display record function is able to display the whole current record, or filter and display the record of members with the same course code.

The development of this data management system had allowed the person in charge to insert and manipulate the records at ease without the needs of papers for members' documentation. It will also allow the person in charge to register the students' data more effectively in a systematic way.

### **4.0 METHODOLOGY**

The data structure elements that had been employed in the Club Registration System includes built-in data types, derived data types, and user defined data types. The derived data types are built by combining both primary and built-in data types and then associating operations on them.

The built-in data types that are used in the systems are as follows:

- (a) Integer
- (b) Character
- (c) Void

The derived data types that are used in the systems are as follows:

- (d) Array
- (e) Pointer
- (f) Linked List

The user defined data types that are used in the systems are as follows:

- (g) Structure

The main data structure elements that were chosen as the backbone of this system is linked list. This is majorly because linked list is a very flexible and dynamic data structure in which elements can be added to or deleted from anywhere at will. This characteristic meets the needs of the club registration systems especially when a specific member needs to be deleted when the member quits the club. Besides, this feature also allows the programmers to write robust programs which require much less maintenance because the list can be added as long as the memory of the device is still available.

The section below describes each function that is in the system.

#### 4.1 void mainMenu()

```
■ void mainMenu(){  
    puts("----- FSKTM IT CLUB REGISTRATION SYSTEM -----\\n\\n"  
        "\\t    <<< MAIN MENU >>>\\n"  
        " \\t(1) Insert A New Record\\n"  
        " \\t(2) Remove A Record\\n"  
        " \\t(3) Update A Record\\n"  
        " \\t(4) Find Record\\n"  
        " \\t(5) Display Record\\n"  
        " \\t(0) EXIT SYSTEM \\n\\n"  
        "     --- [PRESS TWICE ON SELECTION]   ---\\n");  
}
```

The mainMenu( ) function is displayed during the initial runtime of the source code. This function is being called from the main( ) function and it does not receive parameter nor return value to the caller function. It works as a guideline for the user to further choose the functions that is available within the system.

#### 4.2 void insertRecord( )

```
void insertRecord(nodePtr *headptr, newInfo rinfo){
    nodePtr newptr, prevptr, currptr;
    int registeredmember=0;
    newptr = malloc(sizeof(node));           //dynamic memory allocation

    if(newptr != NULL){
        newptr->record = rinfo;             //insertion of record into new node
        newptr->next = NULL;

        prevptr = NULL;
        currptr = *headptr;

        while(currptr != NULL){           //find the end of list when list is not empty
            prevptr = currptr;
            currptr = currptr->next;
        }

        if(prevptr == NULL){              //linking new node when list is empty
            *headptr = newptr;
        }
        else{
            prevptr->next = newptr;      //linking new node when list is not empty
        }

        currptr = *headptr;               //Club Status Analysis Counter
        while(currptr!=NULL){
            registeredmember += 1;
            currptr = currptr->next;
        }
        puts("____");
        puts("\n!!! NEW MEMBER REGISTERED SUCCESSFULLY !!!\n");
        "\tWelcome to IT Club FSKTM :)");
        puts("____");
        printf("\nCURRENT CLUB REGISTRATION STATUS\n");
        printf("  Current Registration Number\t: %d\n", rinfo.no);
        printf("  Current Registered Member\t: %d\n", registeredmember);
    }
    else{
        printf("No memory available.\n");   //dynamic memory allocation failed
    }
}
```

The `insertRecord()` function is being called from the `main()` function when user input ‘1’ after the `mainMenu()` function is being executed. This function receives 2 parameters, `*headptr` and `rinfo`. `*headptr` is passed to the function by reference while `rinfo` is passed to the function by value. This function does not return value to the caller function.

A variable named `registeredmember` is being declared with integer data type in this function. After the insertion process, the linked list is then read again from `*headptr` to calculate the total number of records exist in the list. The variable `registeredmember` will be added by 1 before shifting the `currptr` to the next node until the `currptr` points to `NULL`. From here, the system is able to display the current successfully registered members to the user. Besides, the user could also get the total number of registrations done in the system through reading the struct member “`no`” from the newest record that had just being inserted to the list.

#### 4.3 void removeRecordMenu( )

```
■ void removeRecordMenu(){
    puts("----- FSCTM IT CLUB REGISTRATION SYSTEM -----\\n\\n"
        "     <<< REMOVE RECORD MENU >>>\\n"
        "     \\t(1) By Matrics Number\\n"
        "     \\t(2) By I/C Number\\n"
        "     \\t(0) Return to Main Menu\\n\\n"
        "     --- [PRESS TWICE ON SELECTION] ---\\n");
}
```

The removeRecordMenu( ) function is being called from the main( ) function when user input '2' after the mainMenu( ) function is being executed. This function does not receive parameter and does not return value to the caller function. It works as a guideline for the user to choose the different ways to find and remove a record.

#### 4.4 void removeRecord( )

```
■ void removeRecord(nodePtr *headptr, nodePtr ptr){
    nodePtr temp, prev, curr;
    if(*headptr == ptr){           //remove record when the record is first in the list
        temp = ptr;
        *headptr = ptr->next;
        free(temp);
    }
    else{                         //remove record when the record is not first in the list
        prev = *headptr;
        curr = (*headptr)->next;
        while(curr!=ptr){         //finding the record to be deleted in the list
            prev = curr;
            curr = curr->next;
        }
        temp = curr;
        prev->next = curr->next;
        free(temp);
    }
    int registeredmember = 0;
    curr = *headptr;             //Club Status Analysis Counter
    while(curr!=NULL){
        registeredmember += 1;
        curr = curr->next;
    }
    puts("_____");
    printf("\\nCURRENT CLUB REGISTRATION STATUS\\n");
    printf("  Current Registered Member\\t: %d\\n", registeredmember);
    return;
}
```

The removeRecord( ) function is being called from the main( ) function after the findByMatric( ) or findByIC( ) returns a value back to the main( ) function. This function receives 2 parameters, *\*headptr* and *ptr*. *\*headptr* is passed to the function by reference while *ptr* is passed to the function by value, whereas the value here is the value returned from the findByMatric( ) or findByIC( ) function. This function does not return value to the caller function.

#### 4.5 void updateRecordMenu( )

```
void updateRecordMenu(){
    puts("----- FSKTM IT CLUB REGISTRATION SYSTEM -----\\n\\n"
        "\\t <<< UPDATE RECORD MENU >>>\\n"
        "\\n **SEARCH THE STUDENT DATA FIRST BEFORE UPDATE**\\n\\n"
        " \\t (1) By Matric Number\\n"
        " \\t (2) By I/C Number\\n"
        " \\t (0) Return to Main Menu\\n\\n"
        " --- [PRESS TWICE ON SELECTION] ---\\n");
}
```

The updateRecordMenu( ) function is being called from the main( ) function when user input '3' after the mainMenu( ) function is being executed. This function does not receive parameter and does not return value to the caller function. It works as a guideline for the user to choose the different ways to find the record before proceeding to update the record.

#### 4.6 void updateRecord()

```
void updateRecord(nodePtr *headptr, nodePtr tempPtr){
    int select;
    char matrics[9];
    char name[20];
    char courseCode[4];
    char icNum[13];
    int currStudyYear;

    system("cls");
    updateRecordMenu();
    printf("____"); //To show the record to be updated
    printf("\\n\\t(1) Matric Number\\t: %s\\n", tempPtr->record.matrics);
    printf("\\t(2) Full Name\\t: %s\\n", tempPtr->record.name);
    printf("\\t(3) Course Code\\t: %s\\n", tempPtr->record.courseCode);
    printf("\\t(4) I/C Number\\t: %s\\n", tempPtr->record.icNum);
    printf("\\t(5) Current Study Year : Year %d\\n", tempPtr->record.currStudyYear);

    printf("\\n\\t Enter the Update Selection >> "); //To choose which data from the record to be updated
    scanf("%d", &select);
    switch(select){
        case 1:
            fflush(stdin);
            printf("Please enter NEW matric number >> ");
            gets(matrics);
            strcpy(tempPtr->record.matrics, matrics);
            break;
        case 2:
            fflush(stdin);
            printf("Please enter NEW name >> ");
            gets(name);
            strcpy(tempPtr->record.name, name);
            break;
        case 3:
            fflush(stdin);
            printf("Please enter NEW course code >> ");
            gets(courseCode);
            strcpy(tempPtr->record.courseCode, courseCode);
            break;
        case 4:
            fflush(stdin);
            printf("Please enter NEW IC number >> ");
            gets(icNum);
            strcpy(tempPtr->record.icNum, icNum);
            break;
        case 5:
            fflush(stdin);
            printf("Please enter NEW current study year >> ");
            scanf("%d", &currStudyYear);
            tempPtr->record.currStudyYear = currStudyYear;
            break;
        default:
            printf("\\nOut of choice limit\\n");
            return;
    }
}
```

The updateRecord( ) function is being called from the main( ) function after the findByMatric( ) or findByIC( ) returns a value back to the main( ) function. This function receives 2 parameters, *\*headptr* and *tempPtr*. *\*headptr* is passed to the function by reference while *tempPtr* is passed to the function by value, whereas the value here is the value returned from the findByMatric( ) or findByIC( ) function. This function does not return value to the caller function.

This function further labelled the chosen data with 1, 2, 3, 4, 5 in order to guide the user to choose the attribute to be updated. The switch case is used to further update the data from the chosen attribute.

#### 4.7 void findRecordMenu( )

```
■ void findRecordMenu(){
    puts("----- FSKTM IT CLUB REGISTRATION SYSTEM -----\\n\\n"
        "\\t <<< FIND RECORD MENU >>>\\n"
        " \\t(1) By Matrics Number\\n"
        " \\t(2) By I/C Number\\n"
        " \\t(0) Return to Main Menu\\n\\n"
        " --- [PRESS TWICE ON SELECTION] ---\\n");
}
```

The findRecordMenu( ) function is being called from the main( ) function when user input ‘4’ after the mainMenu( ) function is being executed. This function does not receive parameter and does not return value to the caller function. It works as a guideline for the user to choose the different ways to find the record.

#### 4.8 nodePtr findByMatric( )

```
■ nodePtr findByMatric(nodePtr ptr, char data[9]){
    int i;
    ■ if(ptr == NULL){                                     //if the list is empty
        printf("\\n\\n\\t {{ List is empty }}\\n");
        printf(" No Club Members had been recorded yet\\n");
        printf(" Unable to search for matric %s\\n\\n", data);
        return;
    }
    ■ else{                                              //if the list is not empty
        while(ptr != NULL){
            if(strcmp(ptr->record.matrics,data)==0){
                puts("\\n Student record found.\\n");
                puts("\\t--- STUDENT DETAILS --- ");
                printf(" Record Number\\t\\t: %d\\n", ptr->record.no);
                printf(" Matric Number\\t\\t: %s\\n", ptr->record.matrics);
                printf(" Full Name\\t\\t: %s\\n", ptr->record.name);
                printf(" Course Code\\t\\t: %s\\n", ptr->record.courseCode);
                printf(" I/C Number\\t\\t: %s\\n", ptr->record.icNum);
                printf(" Current Study Year \\t: Year %d\\n", ptr->record.currStudyYear);
                return ptr;
            }
            ■ else{
                ptr = ptr->next;
            }
        }
        puts("\\n\\n      Sorry, record was not found");
        printf("Club Member of Matric %s does not exist\\n", data);
        getch();
        return NULL;
    }
}
```

The `findByMatric()` function is being called from the `main()` function after the user chooses ‘1’ when the `findRecordMenu()` is being called. This function receives 2 parameters, `ptr` and `data[9]`. Both `ptr` and `data[9]` is passed to the function by value. This function returns value to the main function whereas the value is the location of the node of the data that had been found. If the data had not been found, the function will return NULL.

#### 4.9 nodePtr findByIC( )

```
nodePtr findByIC(nodePtr ptr, char data[13]){
    int i;
    if(ptr == NULL){
        printf("\n\n\t { List is empty }\n");
        printf(" No Club Members had been recorded yet\n");
        printf(" Unable to search for IC %s\n", data);
        return;
    }
    else{
        while(ptr != NULL){
            if(strcmp(ptr->record.icNum, data)==0){
                puts("\nStudent record found.\n");
                puts("\t--- STUDENT DETAILS --- ");
                printf(" Record Number\t:t: %d\n", ptr->record.no);
                printf(" I/C Number\t:t: %s\n", ptr->record.icNum);
                printf(" Full Name\t:t: %s\n", ptr->record.name);
                printf(" Matric Number\t:t: %s\n", ptr->record.matrics);
                printf(" Course Code\t:t: %s\n", ptr->record.courseCode);
                printf(" Current Study Year \t:t: Year %d\n", ptr->record.currStudyYear);
                return ptr;
            }
            else{
                ptr = ptr->next;
            }
        }
        puts("\n\n      Sorry, record was not found\n");
        printf("Club Member of IC %s does not exist\n", data);
        getch();
        return NULL;
    }
}
```

The `findByIC()` function works the same as `findByMatric()` function. It is being called from the `main()` function after the user chooses ‘2’ when the `findRecordMenu()` is being called. This function receives 2 parameters, `ptr` and `data[13]`. Both `ptr` and `data[13]` is passed to the function by value. This function returns value to the main function whereas the value is the location of the node of the data that had been found. If the data had not been found, the function will return NULL.

#### 4.10 void displayMenu( )

```
void displayMenu(){
    puts("----- FSKT M IT CLUB REGISTRATION SYSTEM -----\\n\\n"
    "\\t <<< DISPLAY RECORD MENU >>>\\n"
    " \\t(1) All Record\\n"
    " \\t(2) Group By Course\\n"
    " \\t(0) Return to Main Menu\\n\\n"
    " --- [PRESS TWICE ON SELECTION] ---\\n");
}
```

The `findRecordMenu()` function is being called from the `main()` function when user input ‘5’ after the `mainMenu()` function is being executed. This function does not receive parameter and does not return value to the caller function. It works as a guideline for the user to choose the different ways to display the record.

#### 4.11 void displayAll ()

```
■ void displayAll(nodePtr ptr){  
■     if(ptr == NULL){  
■         printf("\nThe list is empty");  
■     }  
■     while(ptr != NULL){  
■         printf("\nRECORD NUMBER %d\n", ptr->record.no);  
■         printf("\t Matriic Number\t : %s\n",ptr->record.matriics);  
■         printf("\t Student Name \t : %s\n",ptr->record.name);  
■         printf("\t Course Code\t : %s\n",ptr->record.courseCode);  
■         printf("\t I/C Number\t : %s\n",ptr->record.icNum);  
■         printf("\t Current Study Year : %d\n\n",ptr->record.currStudyYear);  
■         ptr = ptr->next;  
■     }  
■     getch();  
■ }
```

The `displayAll()` function is being called from the `main()` function after the user chooses ‘1’ when the `displayMenu()` is being called. This function receives one parameter, `ptr`. `ptr` is passed to the function by value. This function does not return value to the main function. The `ptr` will read the and print the data obtained from the first node and further move to the next node to do the same process, until `ptr` points towards `NULL`.

#### 4.12 void displayByCourse ()

```
■ void displayByCourse(nodePtr ptr, char course[4]){  
■     int counter;  
■     if(ptr == NULL){  
■         printf("The list is empty");  
■     }  
■     while(ptr != NULL){  
■         if(strcmp(ptr->record.courseCode,course) == 0){  
■             printf("\nRECORD NUMBER %d\n", ptr->record.no);  
■             printf("\n\t Matriic Number\t : %s\n",ptr->record.matriics);  
■             printf("\t Student Name \t : %s\n",ptr->record.name);  
■             printf("\t I/C Number\t : %s\n",ptr->record.icNum);  
■             printf("\t Current Study Year : %d\n\n",ptr->record.currStudyYear);  
■             counter++;  
■         }  
■         ptr = ptr->next;  
■     }  
■     if(counter == 0){  
■         printf("\nNo club member is in %s.", course);  
■     }  
■     getch();  
■ }
```

The `displayByCourse()` function is being called from the `main()` function after the user chooses ‘2’ when the `displayMenu()` is being called. This function receives 2 parameters, `ptr` and `course[4]`. Both `ptr` and `course[4]` is passed to the function by value. This function

does not return value to the main function. The function will use the data from *ptr* to compare the value passed to the function, *course[4]*. If the value is not the same, *ptr* will shift to the next node and continue the comparison until *ptr* points to NULL. If *ptr* points to NULL but the record of the same course is still not found, the function will print an output stating that the member of that course code is not found. In contrast, the function will print every set of record with the course searched.

## 4.13 int main( )

```
int main(){
    nodePtr head = NULL;
    newInfo readinfo;
    char courseCodeValid[5][4] = {"BIT", "BIP", "BIW", "BIS", "BIM"};
    int recordcount = 0;
    char matricsearch[9], ICsearch[13];
    while(1){
        char matricsearch[9], ICsearch[13], course[4];
        char choice='\\0',choice2='\\0';
        int i=0, j=0, result=3;
        nodePtr temp=NULL;
        mainMenu();
        getch();
        choice = getch();
        switch(choice){
            case '1':
                recordcount++;
                readinfo.no = recordcount;
                printf("_____\n--- Please enter the NEW MEMBER's details ---\n\n");
                printf("Matrix Number\t : ");
                scanf("%[^\\n]s",&readinfo.matrics);
                fflush(stdin);
                printf("Student Name\t : ");
                scanf("%[^\\n]s",&readinfo.name);
                printf("Course code\t : ");
                scanf("%s",&readinfo.courseCode);
                for(i=0; i<5; i++){
                    if(strcmp(readinfo.courseCode,courseCodeValid[i]) == 0){
                        printf("I/C Number\t : ");
                        scanf("%s",&readinfo.icNum);
                        printf("Current study year : ");
                        scanf("%d",&readinfo.currStudyYear);
                        if(readinfo.currStudyYear > 0 && readinfo.currStudyYear <5){
                            fflush(stdin);
                            insertRecord(&head,readinfo);
                            break;
                        }
                    }
                    else{
                        puts("\n Invalid Study Year\n"
                            "Record is not inserted");
                        recordcount--;
                        break;
                    }
                }
            }
        }
    }
}
```

```

        if(i==4){
            if(strcmp(readinfo.courseCode,courseCodeValid[i]) != 0){
                printf("\nSorry, Registration Failed");
                printf("\nCourse Code must be under FSKTM");
                recordcount--;
                break;
            }
        }
        break;

    case '2':
        do{
            fflush(stdin);
            system("cls");
            removeRecordMenu();
            getch();
            choice2 = getch();
            printf("_____ \n");
            switch(choice2){
                case '1':
                    printf("\n Club Member's Matric to Remove Record >> ");
                    gets(matricsearch);
                    printf("_____ ");
                    temp = findByMatric(head, matricsearch);
                    getch();
                    break;
                case '2':
                    printf("\nClub Member's I/C to Remove Record >> ");
                    gets(ICsearch);
                    printf("_____ ");
                    temp = findByIC(head, ICsearch);
                    getch();
                    break;
                case '0':
                    printf("\n      Returning Back to Main Menu...\n");
                    printf("----- [Press Anything to Proceed] ----- \n");
                    getch();
                    break;
                default:
                    puts("\n\t\t!! ERROR !!\n"
                        "\tPlease choose again from the menu.\n"
                        "\tEnter 0 to Return to Main Menu");
                    getch();
                    break;
            }
            if(temp != NULL && (choice2 == '1' || choice2 == '2')){
                printf("\n---[PRESS ENTER TWICE TO PROCEED DELETION]--- \n");
                getch();
                removeRecord(&head, temp);
                break;
            }
        }while(choice2 != '0');
        break;
    case '3':
        do{
            fflush(stdin);
            system("cls");
            updateRecordMenu();
            getch();
            choice2 = getch();
            printf("_____ ");
            switch(choice2){

```

```

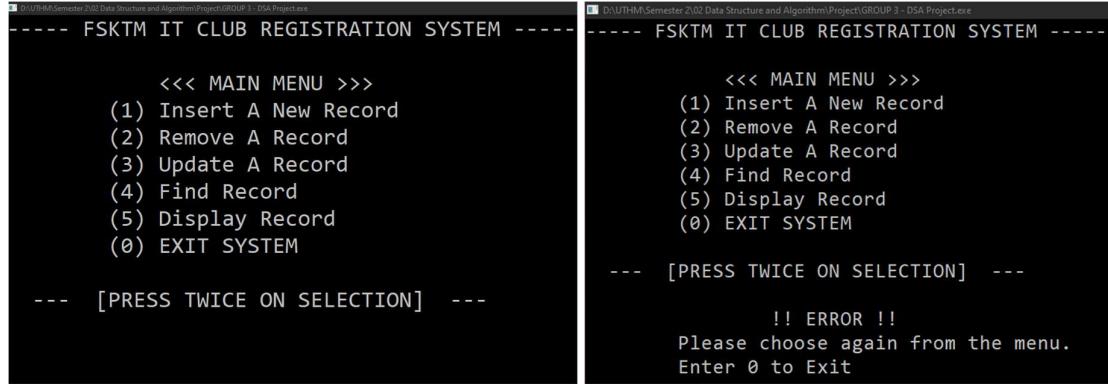
        case '1':
            printf("\nMatric Number of Member to be Updated : ");
            gets(matricsearch);
            printf("_____");
            temp = findByMatric(head, matricsearch);
            break;
        case '2':
            printf("\nIC Number of Member to be Updated : ");
            gets(ICsearch);
            printf("_____");
            temp = findByIC(head, ICsearch);
            break;
        case '0':
            printf("_____");
            printf("\n      Returning Back to Main Menu...\n");
            printf("----- [Press Anything to Proceed] -----");
            break;
        default:
            puts("\n\t\t!! ERROR !!\n"
                "\tPlease choose again from the menu.\n"
                "\tEnter 0 to Return to Main Menu");
            getch();
            break;
    }
    if(temp != NULL && (choice2 == '1' || choice2 == '2')){
        getch();
        printf(" \n---[PRESS ENTER TWICE TO PROCEED UPDATE]---\n");
        updateRecord(&head, temp);
    }
    fflush(stdin);
}while(choice2 != '0');
break;
case '4':
do{
    fflush(stdin);
    system("cls");
    findRecordMenu();
    getch();
    choice2 = getch();
    printf("_____");
    switch(choice2){
        case '1':
            printf("Matric Number to be searched : ");
            gets(matricsearch);
            findByMatric(head, matricsearch);
            getch();
            break;
        case '2':
            printf("IC Number to be searched : ");
            gets(ICsearch);
            findByIC(head, ICsearch);
            getch();
            break;
        case '0':
            printf("\n      Returning Back to Main Menu...\n");
            printf("----- [Press Anything to Proceed] -----");
            break;
        default:
            puts("\t\t !! ERROR !!\n"
                "\t Please choose again from the menu.\n"
                "\t Enter 0 to Return to Main Menu");
            getch();
            break;
    }
}

```

```
        }while(choice2 != '0');
        break;
    case '5':
        do{
            fflush(stdin);
            system("cls");
            displayMenu();
            getch();
            choice2 = getch();
            printf("_____");
            switch(choice2){
                case '1':
                    displayAll(head);
                    break;
                case '2':
                    printf("\n\tPlease enter the course code: ");
                    scanf("%s",&course);
                    for(i=0; i<6; i++){
                        if(strcmp(course,courseCodeValid[i]) == 0){
                            displayByCourse(head,course);
                            break;
                        }
                        if(i==5){
                            if(strcmp(course,courseCodeValid[i]) != 0){
                                printf("\nSorry, Invalid Course Code");
                                printf("\nCourse Code must be under FSKTM");
                                break;
                            }
                        }
                    }
                case '0':
                    printf("\n      Returning Back to Main Menu...\n");
                    printf("----- [Press Anything to Proceed] -----");
                    break;
                default:
                    puts("\t\t !! ERROR !!\n"
                        "\t Please choose again from the menu.\n"
                        "\t Enter 0 to Return to Main Menu");
                    getch();
                    break;
            }
        }while(choice2 != '0');
        break;
    case '0':
        puts("Thank you for using this service\n"
            "Goodbye!");
        exit(0);
    default:
        puts("\t\t !! ERROR !!\n"
            "\tPlease choose again from the menu.\n"
            "\tEnter 0 to Exit");
        break;
    }
    fflush(stdin);
    getch();
    system("cls");
}
```

## 5.0 RESULT

This section will illustrate the process of using every single function in the system. Input data validation will be highlighted in each part too.



```
D:\UTHM\Semester 2\02 Data Structure and Algorithm\Project\GROUP 3 - DSA Project.exe
----- FSKTM IT CLUB REGISTRATION SYSTEM -----
<<< MAIN MENU >>>
(1) Insert A New Record
(2) Remove A Record
(3) Update A Record
(4) Find Record
(5) Display Record
(0) EXIT SYSTEM
--- [PRESS TWICE ON SELECTION] ---

D:\UTHM\Semester 2\02 Data Structure and Algorithm\Project\GROUP 3 - DSA Project.exe
----- FSKTM IT CLUB REGISTRATION SYSTEM -----
<<< MAIN MENU >>>
(1) Insert A New Record
(2) Remove A Record
(3) Update A Record
(4) Find Record
(5) Display Record
(0) EXIT SYSTEM
--- [PRESS TWICE ON SELECTION] ---
!! ERROR !!
Please choose again from the menu.
Enter 0 to Exit
```

Figure 1 : Main Menu and Main Menu Input Validation

During the initial runtime, the main menu will first be shown.

Input data validation exist in the main menu selection. The user can only choose and select from 1 to 5, and 0 to exit. If user select any other symbol, number, or character other than this 6, an error message will be shown, followed by allowing user to reselect their choice from the main menu.

The following are the results of choosing each selection from the main menu.

### 5.1 Insert A New Record



```
D:\UTHM\Semester 2\02 Data Structure and Algorithm\Project\GROUP 3 - DSA Project.exe
----- FSKTM IT CLUB REGISTRATION SYSTEM -----
<<< MAIN MENU >>>
(1) Insert A New Record
(2) Remove A Record
(3) Update A Record
(4) Find Record
(5) Display Record
(0) EXIT SYSTEM
--- [PRESS TWICE ON SELECTION] ---

--- Please enter the NEW MEMBER's details ---

Matrix Number      : AI210360
Student Name       : CHIA SZE YEE
Course code        : BIE

Sorry, Registration Failed
Course Code must be under FSKTM
```

```

----- FSKTM IT CLUB REGISTRATION SYSTEM -----

      <<< MAIN MENU >>>
      (1) Insert A New Record
      (2) Remove A Record
      (3) Update A Record
      (4) Find Record
      (5) Display Record
      (0) EXIT SYSTEM

      --- [PRESS TWICE ON SELECTION] ---

----- Please enter the NEW MEMBER's details ---

Matrix Number      : AI210360
Student Name       : CHIA SZE YEE
Course code        : BIT
I/C Number         : 010520010966
Current study year : 6

      Invalid Study Year
      Record is not inserted

```

*Figure 2 : Insert A New Record with Invalid Course Code and Study Year*

To insert a new record, the number ‘1’ is pressed twice. Then the instructions to input the information needed for the registration is shown line by line. There are 6 details to be filled up to complete the registration.

Input data validation exist during the insertion of course code and study year. If the course code is not under FS KTM and study year is not between 1 and 4, the registration will be cancelled, and record will not be inserted into the linked list.

```

----- FSKTM IT CLUB REGISTRATION SYSTEM -----

      <<< MAIN MENU >>>
      (1) Insert A New Record
      (2) Remove A Record
      (3) Update A Record
      (4) Find Record
      (5) Display Record
      (0) EXIT SYSTEM

      --- [PRESS TWICE ON SELECTION] ---

----- Please enter the NEW MEMBER's details ---

Matrix Number      : AI210360
Student Name       : CHIA SZE YEE
Course code        : BIW
I/C Number         : 010520010966
Current study year : 1

      !!! NEW MEMBER REGISTERED SUCCESSFULLY !!!
      Welcome to IT Club FS KTM :)

----- Please enter the NEW MEMBER's details ---

Matrix Number      : AI210411
Student Name       : KOR WEI YAN
Course code        : BIT
I/C Number         : 011206100318
Current study year : 2

      !!! NEW MEMBER REGISTERED SUCCESSFULLY !!!
      Welcome to IT Club FS KTM :)

----- CURRENT CLUB REGISTRATION STATUS -----
      Current Registration Number : 1
      Current Registered Member   : 1

----- CURRENT CLUB REGISTRATION STATUS -----
      Current Registration Number : 2
      Current Registered Member   : 2

```

*Figure 3 : Successful insertion of records*

When the new member's registration details are all inserted correctly, a message indicating "new member had been registered successfully" and a welcome message will be shown on the output screen. Then an analysis current club registration status will be shown after every successful record insertion.

## 5.2 Remove A Record

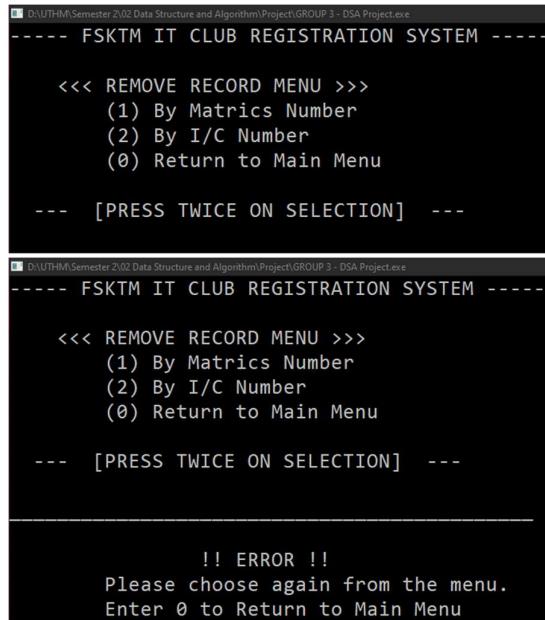


Figure 4 : Remove Record Menu and Remove Record Menu Data Validation

To remove a record, the number '2' is pressed twice from the main menu. Then, this remove record menu shown in Figure 4 will be shown on the output screen. This remove record menu is to assist the system and user to search the set of record to be deleted before proceeding to the deletion process.

Input data validation exist in the remove record menu selection. The user can only choose and select 1 or 2, or 0 to exit. If user select any other symbol, number, or character other than this 3, an error message will be shown, followed by allowing user to reselect their choice from the remove record menu.

```

D:\UTHM\Semester 2\02 Data Structure and Algorithm Project\GROUP 3 - DSA Project.exe
----- FSSTM IT CLUB REGISTRATION SYSTEM -----
<<< REMOVE RECORD MENU >>>
(1) By Matrics Number
(2) By I/C Number
(0) Return to Main Menu

--- [PRESS TWICE ON SELECTION] ---

Club Member's Matric to Remove Record >> CI666666

Sorry, record was not found
Club Member of Matric CI666666 does not exist

D:\UTHM\Semester 2\02 Data Structure and Algorithm Project\GROUP 3 - DSA Project.exe
----- FSSTM IT CLUB REGISTRATION SYSTEM -----
<<< REMOVE RECORD MENU >>>
(1) By Matrics Number
(2) By I/C Number
(0) Return to Main Menu

--- [PRESS TWICE ON SELECTION] ---

Club Member's Matric to Remove Record >> CI123456

Student record found.

--- STUDENT DETAILS ---
Record Number      : 3
Matric Number     : CI123456
Full Name         : ALI BIN ABU
Course Code       : BIW
I/C Number        : 010101101234
Current Study Year : Year 3

---[PRESS ENTER TWICE TO PROCEED DELETION]---

```

*Figure 5 : Fail and Success Find Record by Matric Number Before Removing Record*

‘1’ is being pressed twice from the remove record menu in order to search a record that is need to be deleted through matric number. Then, an instruction is shown on the output screen to ask the user to insert the matric number of the record to be removed. After inserting the matric number, the system will compare each records’ matric number with the value that had been inserted by the user. If no record has matching matric number with the value inputted by the user, no record will be deleted. Otherwise, the system will show the whole record to the output screen and proceed with the deletion process.

```

D:\UTHM\Semester 2\02 Data Structure and Algorithm\Project\GROUP 3 - DSA Project.exe
----- FSKT M IT CLUB REGISTRATION SYSTEM -----
<<< REMOVE RECORD MENU >>>
(1) By Matrics Number
(2) By I/C Number
(0) Return to Main Menu

--- [PRESS TWICE ON SELECTION] ---

Club Member's I/C to Remove Record >> 011206010987
Student record found.

--- STUDENT DETAILS ---
Record Number      : 4
I/C Number         : 011206010987
Full Name          : CHAI WEI QUAN
Matric Number      : C1098765
Course Code         : BIT
Current Study Year : Year 4

---[PRESS ENTER TWICE TO PROCEED DELETION]---

CURRENT CLUB REGISTRATION STATUS
Current Registered Member : 2

Club Member's I/C to Remove Record >> 01010101010101
Sorry, record was not found
Club Member of IC 010101010101 does not exist

```

Figure 6 : Fail and Success Find Record by I/C Number Before Removing Record

‘2’ is being pressed twice from the remove record menu in order to search a record that is need to be deleted through I/C number. Then, an instruction is shown on the output screen to ask the user to insert the I/C number of the record to be removed. After inserting the I/C number, the system will compare each records’ I/C number with the value that had been inserted by the user. If no record has matching I/C number with the value inputted by the user, no record will be deleted. Otherwise, the system will show the whole record to the output screen and proceed with the deletion process. The number of current club’s registered member will be shown after every removal of members from the system.

```

D:\UTHM\Semester 2\02 Data Structure and Algorithm\Project\GROUP 3 - DSA Project.exe
----- FSKT M IT CLUB REGISTRATION SYSTEM -----
<<< REMOVE RECORD MENU >>>
(1) By Matrics Number
(2) By I/C Number
(0) Return to Main Menu

--- [PRESS TWICE ON SELECTION] ---

Returning Back to Main Menu...
----- [Press Anything to Proceed] -----

```

Figure 7 : Return to Main Menu from Remove Record Menu

‘0’ is being pressed twice from the remove record menu to return to the main menu to access other function.

### 5.3 Update A Record

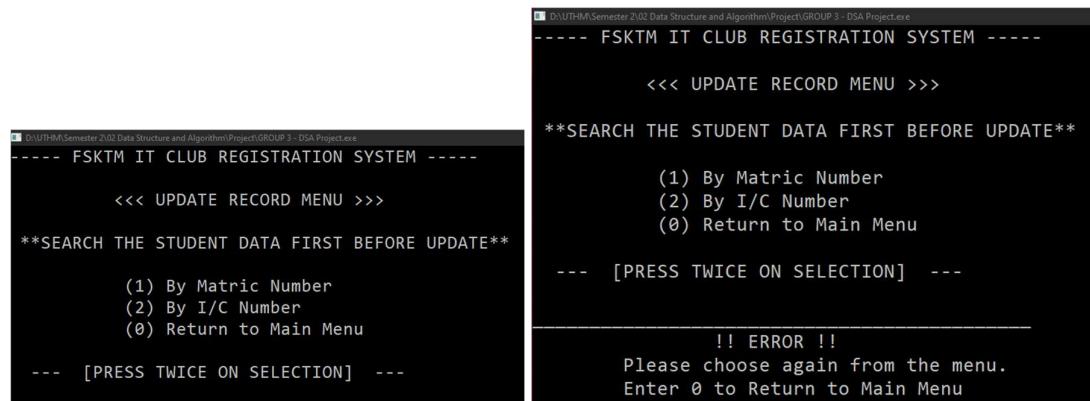


Figure 8 : Update Record Menu and Update Record Menu Data Validation

To update a record, the number ‘3’ is pressed twice from the main menu. Then, this update record menu shown in Figure 8 will be shown on the output screen. This update record menu is to assist the system and user to search the set of record to be updated before proceeding to the updating process.

Input data validation exist in the update record menu selection. The user can only choose and select 1 or 2, or 0 to exit. If user select any other symbol, number, or character other than this 3, an error message will be shown, followed by allowing user to reselect their choice from the update record menu.

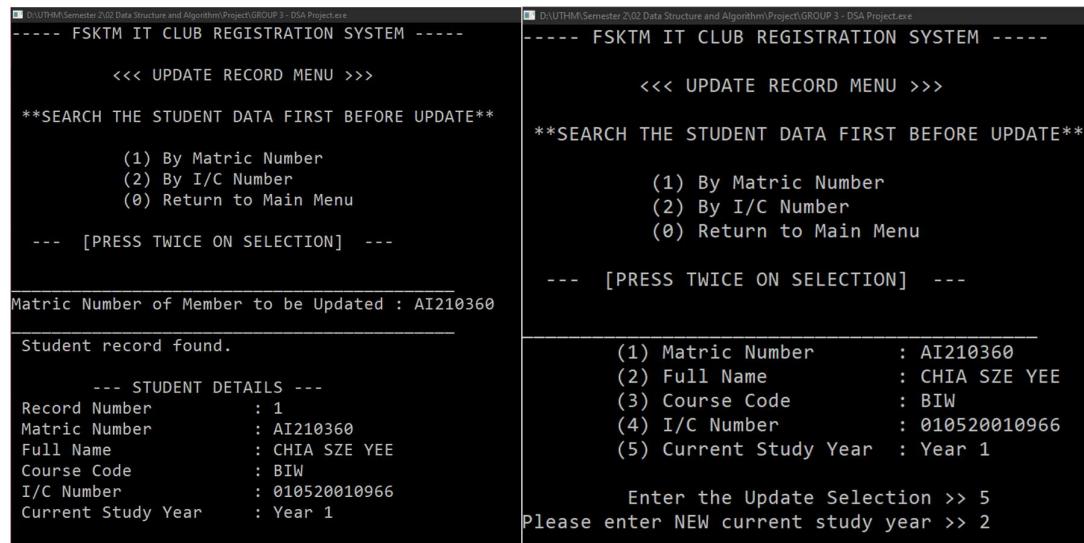


Figure 9 : Update Record Matric Number Search and Update Record

To search the record to be updated through matric number, ‘1’ is pressed twice from the update record menu. Then insert the matric number to search the record. When the record is found, further updating process is done by first labelling each attribute with numbers.

Then, based on the numbers chosen by the user, the updating process is done on the specific attribute of the searched record data set. It works the similarly with the second selection from the update record menu which is shown in Figure 10 below. This allows user to be more flexible in searching the specific record.

```

D:\UTHM\Semester 2\02 Data Structure and Algorithm\Project\GROUP 3 - DSA Project.exe
----- FS KTM IT CLUB REGISTRATION SYSTEM -----
<<< UPDATE RECORD MENU >>>
**SEARCH THE STUDENT DATA FIRST BEFORE UPDATE**
(1) By Matric Number
(2) By I/C Number
(0) Return to Main Menu
--- [PRESS TWICE ON SELECTION] ---

IC Number of Member to be Updated : 011206100318
Student record found.

--- STUDENT DETAILS ---
Record Number      : 2
I/C Number         : 011206100318
Full Name          : KOR WEI YAN
Matric Number      : AI210411
Course Code         : BIT
Current Study Year : Year 2

D:\UTHM\Semester 2\02 Data Structure and Algorithm\Project\GROUP 3 - DSA Project.exe
----- FS KTM IT CLUB REGISTRATION SYSTEM -----
<<< UPDATE RECORD MENU >>>
**SEARCH THE STUDENT DATA FIRST BEFORE UPDATE**
(1) By Matric Number
(2) By I/C Number
(0) Return to Main Menu
--- [PRESS TWICE ON SELECTION] ---

(1) Matric Number      : AI210411
(2) Full Name           : KOR WEI YAN
(3) Course Code          : BIT
(4) I/C Number            : 011206100318
(5) Current Study Year   : Year 2

Enter the Update Selection >> 3
Please enter NEW course code >> BIW

```

Figure 10 : Update Record I/C Number Search and Update Record

#### 5.4 Find A Record

```

D:\UTHM\Semester 2\02 Data Structure and Algorithm\Project\GROUP 3 - DSA Project.exe
----- FS KTM IT CLUB REGISTRATION SYSTEM -----
<<< FIND RECORD MENU >>>
(1) By Matrics Number
(2) By I/C Number
(0) Return to Main Menu
--- [PRESS TWICE ON SELECTION] ---

D:\UTHM\Semester 2\02 Data Structure and Algorithm\Project\GROUP 3 - DSA Project.exe
----- FS KTM IT CLUB REGISTRATION SYSTEM -----
<<< FIND RECORD MENU >>>
(1) By Matrics Number
(2) By I/C Number
(0) Return to Main Menu
--- [PRESS TWICE ON SELECTION] ---

!! ERROR !!
Please choose again from the menu.
Enter 0 to Return to Main Menu

```

Figure 11 : Find Record Menu and Find Record Menu Data Validation

To find a record, the number ‘4’ is pressed twice from the main menu. Then, this find record menu shown in Figure 11 will be shown on the output screen. This menu is to assist the system and user to search the set of record that had been inserted into the system.

Input data validation exist in the find record menu selection. The user can only choose and select 1 or 2, or 0 to exit. If user select any other symbol, number, or character other than this 3, an error message will be shown, followed by allowing user to reselect their choice from the find record menu.

```

D:\UTHM\Semester 2\02 Data Structure and Algorithm\Project\GROUP 3 - DSA Project.exe
----- FSIT CLUB REGISTRATION SYSTEM -----
<<< FIND RECORD MENU >>>
(1) By Matrics Number
(2) By I/C Number
(0) Return to Main Menu

--- [PRESS TWICE ON SELECTION] ---

Matric Number to be searched : AI210360
Student record found.

--- STUDENT DETAILS ---
Record Number      : 1
Matric Number     : AI210360
Full Name          : CHIA SZE YEE
Course Code        : BIW
I/C Number         : 010520010966
Current Study Year : Year 2

D:\UTHM\Semester 2\02 Data Structure and Algorithm\Project\GROUP 3 - DSA Project.exe
----- FSIT CLUB REGISTRATION SYSTEM -----
<<< FIND RECORD MENU >>>
(1) By Matrics Number
(2) By I/C Number
(0) Return to Main Menu

--- [PRESS TWICE ON SELECTION] ---

IC Number to be searched : 011206100318
Student record found.

--- STUDENT DETAILS ---
Record Number      : 2
I/C Number         : 011206100318
Full Name          : KOR WEI YAN
Matric Number     : AI210411
Course Code        : BIW
Current Study Year : Year 2

```

Figure 12 : Find Record by Matric Number and I/C Number

To search the record by matric number, ‘1’ is pressed twice; while to search the record by I/C number, ‘2’ is pressed twice. Then insert the matric or I/C number in the given field. If the matric number or I/C number matches any of the records data, the student details will show up, or else, a message stating the record with the specific I/C or matric is not found. Then, ‘0’ is pressed twice to return back to main menu.

## 5.5 Display Record

```

D:\UTHM\Semester 2\02 Data Structure and Algorithm\Project\GROUP 3 - DSA Project.exe
----- FSIT CLUB REGISTRATION SYSTEM -----
<<< DISPLAY RECORD MENU >>>
(1) All Record
(2) Group By Course
(0) Return to Main Menu

--- [PRESS TWICE ON SELECTION] ---

D:\UTHM\Semester 2\02 Data Structure and Algorithm\Project\GROUP 3 - DSA Project.exe
----- FSIT CLUB REGISTRATION SYSTEM -----
<<< DISPLAY RECORD MENU >>>
(1) All Record
(2) Group By Course
(0) Return to Main Menu

--- [PRESS TWICE ON SELECTION] ---

!! ERROR !!
Please choose again from the menu.
Enter 0 to Return to Main Menu

```

Figure 13 : Display Record Menu and Display Record Menu Data Validation

To find a record, the number ‘5’ is pressed twice from the main menu. Then, this display record menu shown in Figure 13 will be shown on the output screen. This menu is to assist the system and user to search the set of record that had been inserted into the system.

Input data validation exist in the find record menu selection. The user can only choose and select 1 or 2, or 0 to exit. If user select any other symbol, number, or character other than this 3, an error message will be shown, followed by allowing user to reselect their choice from the display record menu.

The figure consists of two side-by-side screenshots of a computer terminal window titled "FSKTM IT CLUB REGISTRATION SYSTEM".

**Left Screenshot (DISPLAY RECORD MENU):**

- Header: "----- FS KTM IT CLUB REGISTRATION SYSTEM -----"
- Section: "<<< DISPLAY RECORD MENU >>>"
- Options:
  - (1) All Record
  - (2) Group By Course
  - (0) Return to Main Menu
- Text: "--- [PRESS TWICE ON SELECTION] ---"
- Section: "RECORD NUMBER 1"
 

Matric Number	:	AI210360
Student Name	:	CHIA SZE YEE
Course Code	:	BIW
I/C Number	:	010520010966
Current Study Year	:	2
- Section: "RECORD NUMBER 2"
 

Matric Number	:	AI210411
Student Name	:	KOR WEI YAN
Course Code	:	BIW
I/C Number	:	011206100318
Current Study Year	:	2

**Right Screenshot (DISPLAY RECORD MENU AND DATA VALIDATION):**

- Header: "----- FS KTM IT CLUB REGISTRATION SYSTEM -----"
- Section: "<<< DISPLAY RECORD MENU >>>"
- Options:
  - (1) All Record
  - (2) Group By Course
  - (0) Return to Main Menu
- Text: "--- [PRESS TWICE ON SELECTION] ---"
- Text: "Please enter the course code: BIW"
- Section: "RECORD NUMBER 1"
 

Matric Number	:	AI210360
Student Name	:	CHIA SZE YEE
I/C Number	:	010520010966
Current Study Year	:	2
- Section: "RECORD NUMBER 2"
 

Matric Number	:	AI210411
Student Name	:	KOR WEI YAN
I/C Number	:	011206100318
Current Study Year	:	2

Figure 14 : Display Record Menu and Display Record Menu Data Validation

To display all the record, ‘1’ is pressed twice; while to display record with the same course code, ‘2’ is pressed twice. When 1 is chosen, all the record stored in the linked list is displayed on the output screen. On the other hand, when 2 is chosen, user is required to insert the name of the course code of the records that the user wanted to search. Then, ‘0’ is pressed twice to return back to main menu.

## 6.0 CONCLUSIONS

With a few ideas and insights on the fulfilment of overall objectives, improvements, benefits, and weaknesses of the system, the findings may be used throughout the development of the Information Technology Club Member Registration System for IT Club FSKTM. Additionally, recommendations for future improvements to the system are noted. Based on the results of the testing of the Information Technology Club Registration System, the overall target performance is satisfied. All of the modules created for this system operate effectively and meet the project's goals and objectives. The goal of the system development is to design and construct an information technology club registration system to replace the manual system now in use. In addition, it is to develop systems that can store a more systematic data.

