Green University of Bangladesh
Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
Semester: (Spring, Year:2021), B.Sc. in CSE (Day/Eve)

Course Title:CSE Lab
Course Code: CSE 106 Section:DD

Lab Project Name:ALBUM MANAGEMENT SYSTEM

Student Details

Name	ID
NASIRUDDIN	213902043

Submission Date: 9/10/2022
Course Teacher's Name: Md. Sultanul Islam Ovi

[For Teachers use only: Don't Write Anything inside this box]

Lab Project Status

Marks: …… **Signature: .....**

Comments: ..... **Date: .....**

Table of Contents

Chapter 1

Introduction

1.1 Introduction

Album management system is specially designed for purpose of adding the music album records in a library. The system elaborates the basic concept for storing and generating the album's detail. Here in this system, staff can sign up as a system admin. He/she can have full access to the system for maintaining records.

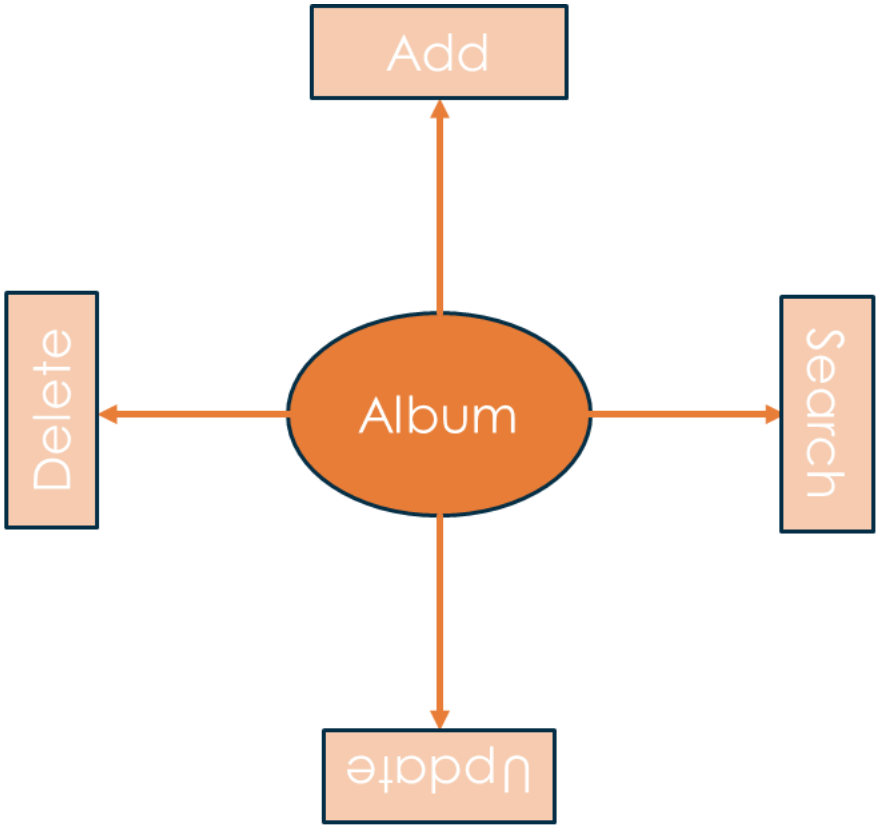
1.2 Design Goals/Objective

- Adding album records.
- Deleting
- Updating records.
- Viewing records

Chapter 2

Design/Development/Implementation of the Project

2.1 Front Screen Design



2.2 PSEUDO CODE

HEADER FILES & VARIABLES

```
#include<stdlib.h>
#include<string.h>
#include<stdio.h>
#define Size 100
```

```
struct Album
{
   int seriulNumber;
   char name[Size];
   char artistName[Size];
   int  price;
   struct Album *next;

}* album;



//MAIN FUNCTION

int main()
{
   album = NULL;
   int choice;
   char name[Size];
   char artistName[Size];
   int seriulNumber;
   int  p;
   printf("\n");
   printf("\n                o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o
");
   printf("\n                *                                   *    ");
   printf("\n                *                                   *    ");
   printf("\n                *            WELCOME TO                 *    ");
   printf("\n                *                                   *    ");
   printf("\n                *          | ALBUM MANAGEMENT |           *    ");
   printf("\n                *                                   *    ");
   printf("\n                *                                   *    ");
   printf("\n                *                                   *    ");
   printf("\n                *                                   *    ");
   printf("\n                *                                   *    ");
   printf("\n                o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o*o
");
   printf("\n\n");
   printf("\t\t\t 1. To insert Album details\n");
   printf("\t\t\t 2. To search for Album details\n");
   printf("\t\t\t 3. To delete Album details\n");
   printf("\t\t\t 4. To update Album details\n");
```

```
printf("\t\t\t 5. To display all Album details");

do
{
   printf("\nEnter Choice: ");
   scanf("%d", &choice);
   switch (choice)
   {
   case 1:
      printf("\n\nEnter Serial number: ");
      scanf("%d", &seriulNumber);
      printf("Enter name: ");
      scanf("%s", name);
      printf("Enter Artist Name: ");
      scanf("%s", artistName);
      printf("Enter price: ");
      scanf("%d", &p);
      insert(seriulNumber,name,artistName,p);
      break;

   case 2:
      printf("Enter Serial Number to search: ");
      scanf("%d", &seriulNumber);
      search(seriulNumber);
      break;

   case 3:
      printf("Enter Serial Number to delete: ");
      scanf("%d", &seriulNumber);
      Delete(seriulNumber);
      break;

   case 4:
      printf("Enter Serial Number to update: ");
      scanf("%d", &seriulNumber);
      update(seriulNumber);
      break;
   case 5:
      display();
      break;
   default:
      printf("Wrong Option\n");
```

```
        }

    }
    while (choice != 0);
}
```

//FUNCTION FOR BINARY SEARCH

```
struct Album* middle(struct Album* start,struct Album* last)
{
    if (start == NULL)
        return NULL;

    struct Album* slow = start;
    struct Album* fast = start -> next;

    while (fast != last)
    {
        fast = fast -> next;
        if (fast != last)
        {
            slow = slow -> next;
            fast = fast -> next;
        }
    }

    return slow;
}
struct Album* binarySearch(struct Album *album, int seriulNumber)
{
    struct Album* start = album;
    struct Album* last = NULL;

    do
    {
        struct Album* mid = middle(start, last);

        if (mid == NULL)
            return NULL;
```

```
        if (mid -> seriulNumber == seriulNumber)
            return mid;

        else if (mid -> seriulNumber < seriulNumber)
            start = mid -> next;

        else
            last = mid;

    }
    while (last == NULL ||
            last != start);

    return NULL;
}
void insert(int seriulNumber, char* name, char* artistName, int  price)
{
    struct Album * Album = (struct Album *) malloc(sizeof(struct Album));
    Album->seriulNumber = seriulNumber;
    strcpy(Album->name, name);
    strcpy(Album->artistName, artistName);
    Album->price = price;
    Album->next = NULL;

    if(album==NULL)
    {
        album = Album;

    }
    else
    {
        Album->next = album;
        album = Album;
    }

}
void search(int seriulNumber)
{
    struct Album * temp = album;

    if(binarySearch(album,seriulNumber)==NULL)
```

```
    {
        printf("Data not found");
    }
    else
    {
        printf("Serial Number: %d\n", temp->seriulNumber);
        printf("Name: %s\n", temp->name);
        printf("artistName: %s\n", temp->artistName);
        printf("price: %d\n", temp->price);
        return;
    }

}
```

//FUNCTION FOR UPDATE ALBUM DETAILS

```
void update(int seriulNumber)
{

    struct Album * temp = album;
    while(temp!=NULL)
    {
        if(temp->seriulNumber==seriulNumber)
        {
            printf("Record with Serial Number %d Found !!!\n", seriulNumber);
            printf("Enter new name: ");
            scanf("%s", temp->name);
            printf("Enter new Artist Name number: ");
            scanf("%s", temp->artistName);
            printf("Enter new price: ");
            scanf("%d",&temp->price);
            printf("Update Successful!!!\n");
            return;
        }
        temp = temp->next;

    }
    printf("Album with Serial Number %d is not found !!!\n", seriulNumber);
}
```

```
//FUNCTION FOR DELETE


void Delete(int seriulNumber)
{
   struct Album * temp1 = album;
   struct Album * temp2 = album;
   while(temp1!=NULL)
   {

      if(temp1->seriulNumber==seriulNumber)
      {

         printf("Album with this Serial Number %d Found !!!\n", seriulNumber);

         if(temp1==temp2)
         {
            album = album->next;
            free(temp1);
         }
         else
         {

            temp2->next = temp1->next;
            free(temp1);
         }

         printf("Album Successfully Deleted !!!\n");
         return;

      }
      temp2 = temp1;
      temp1 = temp1->next;

   }
   printf("Album with this Serial Number %d is not found !!!\n", seriulNumber);

}
```

```
//FUNCTION FOR DISPLAY

void display()
{
   struct Album * temp = album;
   while(temp!=NULL)
   {
      printf("List of Albums");
      printf("Serial Number: %d\n", temp->serialNumber);
      printf("Name: %s\n", temp->name);
      printf("Artist Name: %s\n", temp->artistName);
      printf("\nPrice: %d\n\n",temp->price);
      temp = temp->next;

   }
}
```

```
//FUNCTION FOR SORTING


void sort()
{
   struct Album *curNode,*nextNode;
   curNode = album;
   while(curNode!=0)
   {
      nextNode = curNode->next;

      while(nextNode!=0)
      {
         if(curNode->serialNumber>nextNode->serialNumber)
         {
            swap1(&curNode->serialNumber, &nextNode->serialNumber);
            swap2(curNode->name, nextNode->name);
            swap2(curNode->artistName, nextNode->artistName);
            swap3(&curNode->price, &nextNode->price);
```

```
            }
            nextNode = nextNode->next;
        }
        curNode = curNode->next;
    }
    printf("\nAlbum is Now Sorted\n");
}
```

//FUNCTION OF SWAP

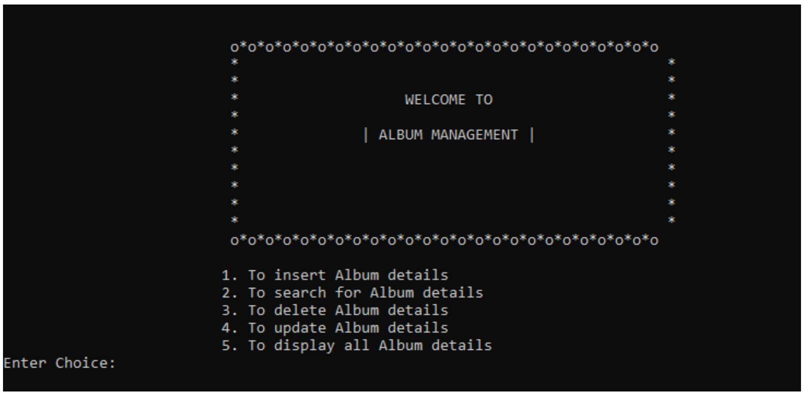
```
void swap1( int *a, int *b)
{
    int temp;
    temp  = *a;
    *a = *b;
    *b = temp;
}


void swap3(float *x, float *y)
{
    float temp;
    temp  = *x;
    *x = *y;
    *y = temp;
}


void swap2(char *str1, char *str2)
{
    char *temp = (char *)malloc((strlen(str1) + 1) * sizeof(char));
    strcpy(temp, str1);
    strcpy(str1, str2);
    strcpy(str2, temp);
    free(temp);
}
```

Chapter 3

Performance Evaluation

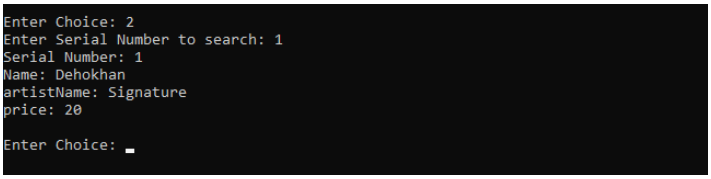
3.1 OUTPUT

HOME SCREEN



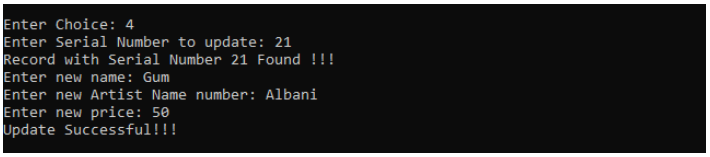
INSERT

SEARCH



DELETE

UPDATE



DISPLAY

3.2 Results and Discussions

Analysis and Outcome

By using student management system we can easily insert a students information,we can search the inserted student information whenever we want ,if we have given the wrong information then we change it and update it,we can display all the information,if the student leaves then we can delete those informations

Chapter 4

Conclusion

Album Management system is a simple mini project in C, built as a console application without using graphics features. It's just a demonstration of the use of file handling in C language. It has been used to jump from one menu to another within the program. Also, for editing and deleting the recorded items, separate functions have not been used

4.1 Scope of Future Work

Album management system is specially designed for purpose of adding the music album records in a library. The system elaborates the basic concept for storing and generating the album's detail. Here in this system, staff can sign up as a system admin. He/she can have full access to the system for maintaining records.

References

[1] Used google to get the basic structure of my code.

[2] Got the idea for younger brother, frands and me.