Green University of Bangladesh
Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering
Semester: (Spring, Year:2022), B.Sc. in CSE (Day)

Course Title: Data Structures Lab
Course Code: CSE106 Section: DB

LAB PROJECT REPORT

Lab Project Name: Employee Record Management System

Student Details

	Name	ID
1.	Waliyel Hasnat Zaman	212002022

Submission Date : 06-05-2022
Course Teacher's Name : Dr. Nazib Abdun Nasir

[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

The name of my project is “Employee Record Management System”. Basically, it stores the records of the employees working in a company. The mini-project “Employee Record Management System” is a console application using the C/C++ programming language. The GCC compiler was used to compile this project in Code Blocks. You can use this console program to perform basic Employee Record functions such as adding employee’s information, viewing added employee records, searching for employees, and so on This application is based on file handling in C/C++ and as it is similar to a menu based program, I’ve used switch statement here. Here, for the sake of efficiency, I have used a number of file-related function like fopen, fread, fwrite, etc.

1.2 Design Goals/Objective

1. To store records of the employees working in a company or office.

2. To increase efficiency in storing Employee’s data and decrease workload in a company or an office.

Chapter 2

Design/Development/Implementation of the Project

2.1 Section (Code of the project)

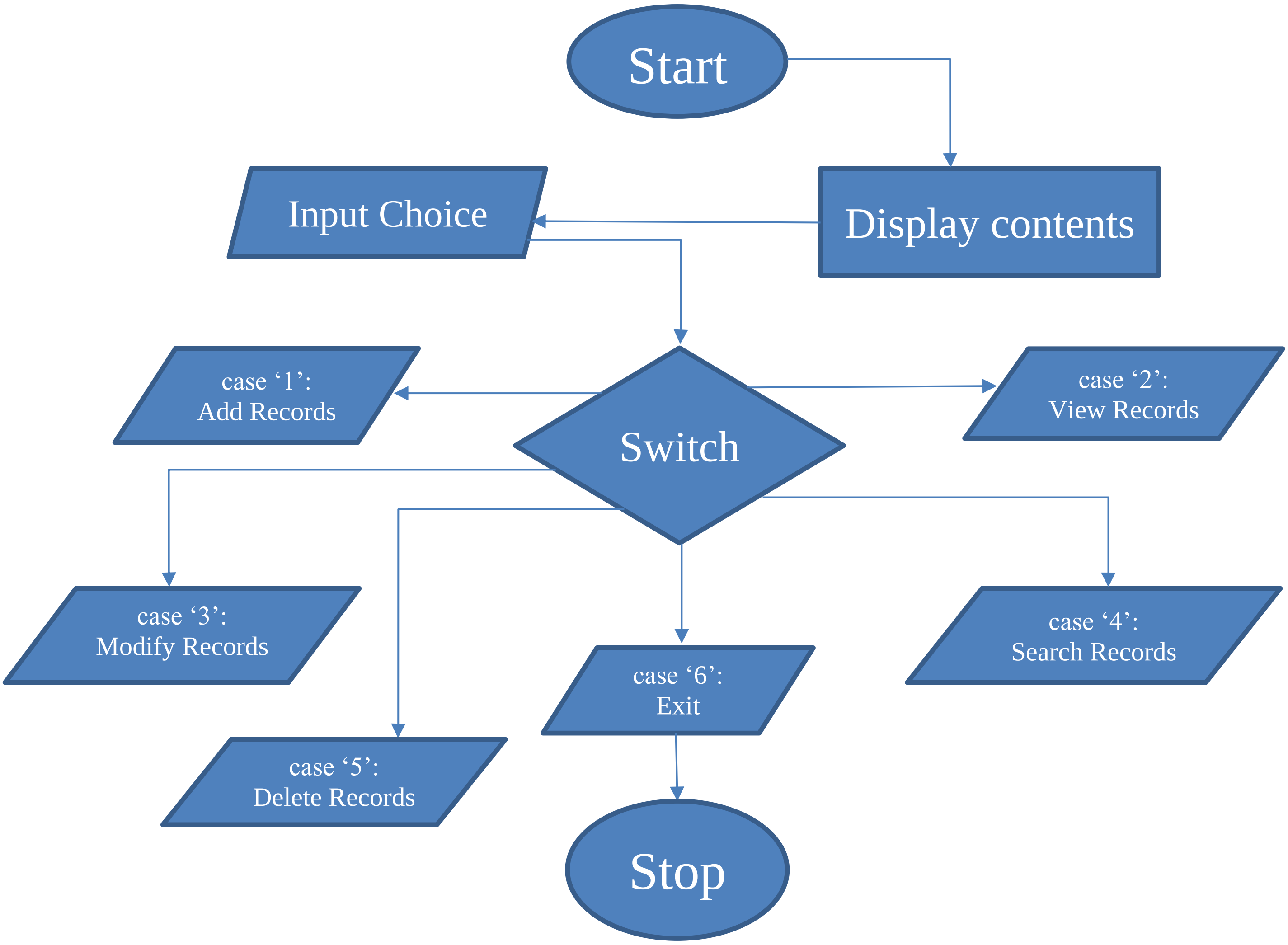
The code of the project is given bellow:

```
///Declared necessary header files.

#include <iostream>
#include <cstdio>
#include <cstring>
#include <cstdlib>
#include <conio.h>
#include <iomanip>
#include<time.h>

/**
*Project Title: Employee Record Management System
*Name: Waliyel Hasnat Zaman
*ID: 212002022
*Sec: DB
*Course Code: CSE 106
*Course Title: Data Structures Lab
*/

using namespace std;


///Declared File Pointers

FILE *fp, *ft; //'ft' represents temporary file pointer.
char another, choice; //Declared variables to store user decisions.
```

```
///Declared structure to represent an employee.

struct employee{

    char ID[10];
    char Name[30];
    int Age;
    long Salary;
};

struct employee e; //Declared structure variable.

char xID[10]; //Variable declared to store the ID of employees for comparison.

long int recsize; //Variable declared to store the size of the structure variable 'e'.


///Delay Function created for mimicking the loading screen.

void delay(unsigned int mseconds){

    clock_t goal = mseconds + clock();
    while (goal > clock());
}


///User defined function to add new records of the employees.

void addrecord(){

    fseek(fp,0,SEEK_END); //Function used to set the cursor position at the end.
    another ='Y';
    while(another == 'Y' || another == 'y'){

        system("cls");
        cout<<"Enter New Employee's ID: ";
        fflush(stdin);
        gets(e.ID);
        cout << "Enter New Employee's Name: ";
        fflush(stdin);
        gets(e.Name);
        cout << "Enter New Employee's Age: ";
        fflush(stdin);
        cin >> e.Age;
        cout << "Enter New Employee's Basic Salary: ";
        cin >> e.Salary;
        fwrite(&e,recsize,1,fp); //Writes the records in the file 'emp.txt'.
        cout << "\n Add Another Record (Y/N) ";
        fflush(stdin);
        another = getchar();
    }
}


///User defined function to view records of the employees.

void viewrecord(){

    system("cls");
    rewind(fp);
    cout << "=== View The Records In The Database ===";
    cout << "\n";

    cout<<"\n\n   ID               NAME                AGE     Salary\n";
    while (fread(&e,recsize,1,fp) == 1) //Function used to read from the file.
        printf("\n%-10s\t%-24s %-2d\t%-10d\n",e.ID, e.Name, e.Age, e.Salary);

    cout << "\n\n";
    system("pause");
}


///User defined function to modify old records of the employees.

void modifyrecord(){

    //Records are modified by comparing the ID given by the user with the IDs saved in the file 'emp.txt'.

    system("cls");
    another = 'Y';
    int flag;
    while (another == 'Y'|| another == 'y'){
        cout << "\nEnter ID Of The Employee To Modify His/Her Records: ";
        fflush(stdin);
        gets(xID);
        flag=0;
        rewind(fp);
        while (fread(&e,recsize,1,fp) == 1){

            fflush(stdin);
            if (strcmp(e.ID, xID)==0){

                cout<<"\n\nModify Employee's Name: ";
                fflush(stdin);
                gets(e.Name);
                cout<<"Modify Employee's Age: ";
                cin>>e.Age;
                cout<<"Modify Employee's Salary: ";
                cin>> e.Salary;
                fseek(fp, - recsize, SEEK_CUR);
                fwrite(&e,recsize,1,fp);
                flag++;
                break;
            }
        }
        if(flag==0)
            cout<<"Record Not Found\n";

        cout << "\nModify Another Record (Y/N): ";
        fflush(stdin);
        another = getchar();
    }
}


///User defined function to search records of the employees.

void searchrecord(){

    //Records are searched by comparing the ID given by the user with the IDs saved in the file 'emp.txt'.

    system("cls");
    char sID[10],CH;
    another='Y';
    while(another=='Y'||another=='y'){
        system("cls");
        cout<<"Enter ID To Search: ";
        fflush(stdin);
        gets(sID);
        rewind(fp);

        cout<<"\n\n   ID              NAME               AGE     Salary\n";
        CH='Y';

        while (fread(&e,recsize,1,fp) == 1){

            if(strcmp(e.ID,sID) == 0)
                    printf("\n%-10s\t%-24s %-1d\t%-10d\n\n\n\n",e.ID, e.Name, e.Age, e.Salary);
            }
        cout<<"\n";
        cout<<"Want To Search Again (Y/N): ";
        fflush(stdin);
        another=getchar();
    }
}


///User defined function to delete records of the employees.

void deleterecord(){

    //Records are deleted by comparing the ID given by the user with the IDs saved in the file
'emp.txt'.

    system("cls");

    another = 'Y';
    while (another == 'Y'|| another == 'y'){

        cout << "\nEnter The ID Of The Employee To Delete: ";
        fflush(stdin);
        cin >> xID;

        ft = fopen("temp.txt", "wb"); //Open 'temp.txt' in Write Binary mode.

        rewind(fp);
        while (fread (&e, recsize,1,fp) == 1)
        if (strcmp(e.ID,xID) != 0)
            fwrite(&e,recsize,1,ft); //If ID given by the user matches with the ID of the file, except the records of that ID the rest will be moved to 'temp.txt'.

        fclose(ft); //File pointer 'ft' closed.
        fclose(fp); //File pointer 'fp' closed.
        remove("emp.txt"); //Remove 'emp.txt'.
        rename("temp.txt","emp.txt"); //Rename 'temp.txt' to 'emp.txt'.

        fp=fopen("emp.txt","rb+"); //Opens 'emp.txt' in Read Binary Plus mode.

        cout<<"Record Deleted Successfully\n";
        cout << "\n\nDelete Another Record (Y/N): ";
        fflush(stdin);
        another = getchar(); //Get another choice from user.
    }
}


///Main function starts from here.

int main(){

    system("cls");

    for(int i=1;i<=50;i++){ //For loading screen.
        system("cls");
        cout<<"\n\n\n\n\n\n\n\n\n\t\t\t\t\t====LOADING==== "<<"\n\n"<<"\t"<<"\t";
        cout<<"\t";
        for(int j=1;j<=i;j++)
            cout<<"|";
        cout<<"\n\n\t\t\t "<<2*i<<"%";
        if( i > 1 && i < 20)
            cout<<"\n\n\t\t\tCreating Temporary files";
        else if( i > 20 && i<40)
                cout<<"\n\n\t\t\tAccessing Main Memory";
        else if(i >40 && i<48)
                cout<<"\n\n\t\t\tAccessing Cache";
        else cout<<"\n\n\t\t\tLoading Complete. Press Enter to Continue ";
        delay(150 - i*3);
      }
    getch();
    recsize = sizeof(e);
    while(1){

        system("cls"); //Clears screen.

        cout << "\n\n\n\n\t\t      ====== EMPLOYEE RECORD MANAGEMENT SYSTEM
======";
        cout <<"\n\n";
        cout << "\n\n";
        cout << "\n \t\t\t\t 1. Add    Records";
        cout << "\n \t\t\t\t 2. View   Records";
        cout << "\n \t\t\t\t 3. Modify Records";
        cout << "\n \t\t\t\t 4. Search Records";
        cout << "\n \t\t\t\t 5. Delete Records";
        cout << "\n \t\t\t\t 6. Exit   Program";
        cout << "\n\n";
        cout << "\t\t\t\t Select Your Choice : ";
        fflush(stdin);
        choice = getche();



        ///Switch statements are used in this program to take choice from user to perform any
operations.

        switch(choice){

        case '1' : //Adds new records.
            fp=fopen("emp.txt","rb+");
            if (fp == NULL){ //If file 'emp.txt' exists the open in rb+ mode else create new 'emp.txt'
fileand open it in wb+ mode.

                fp = fopen("emp.txt","wb+");

                if (fp==NULL){

                    puts("Cannot Open File");
                    return 0;
                }
            }
            addrecord();
            fclose(fp);
            break;
        case '2': //View records.
            fp=fopen("emp.txt","rb+");
            viewrecord(); //Calling user defined function.
            fclose(fp);
            break;

        case '3' : //Modify old records.
            fp=fopen("emp.txt","rb+");
            modifyrecord();
            fclose(fp);
            break;

        case '4' : //Search records.
            fp=fopen("emp.txt","rb+");
            searchrecord();
            fclose(fp);
            break;

        case '5': //Delete records.
            fp=fopen("emp.txt","rb+");
            deleterecord();
            fclose(fp);
            break;

        case '6': //Close/Exits the program.
            fclose(fp);
            cout << "\n\n\n\t\t\t\t     THANK YOU!\n\n";
            exit(0);
        }
    }

    system("pause");
    return 0;
}
```

Note: In the code, I’ve added multiple lines of comments to specify the functions of each section of the code.

2.2 Section (Structure of the Program)



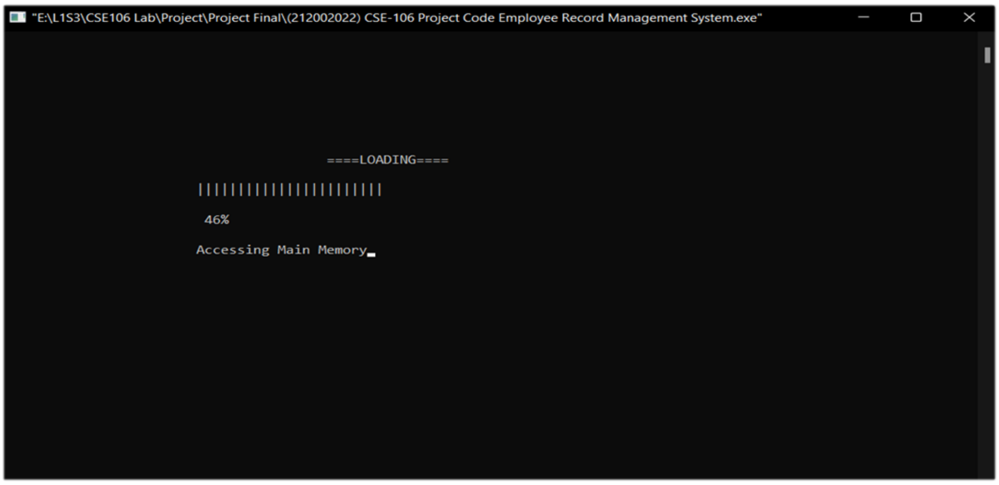
*****The program will execute following the above stucture.**

Chapter 3

Performance Evaluation

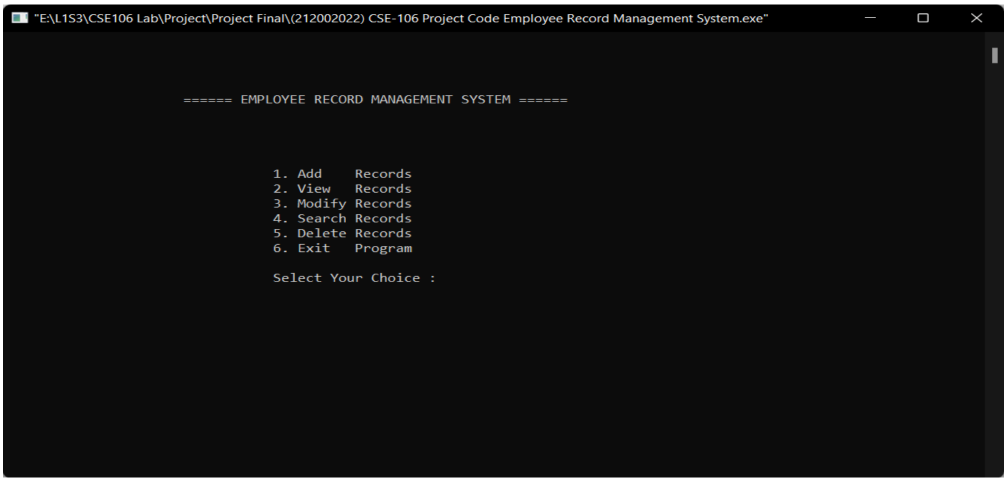
3.1 Simulation Environment/ Simulation Procedure

When we execute the program, a loading screen will appear and then it will display 6 menus and asks for a menu number which user wants to visit. By inputting the preferred choice, a user can add, view, modify, search, delete records or simply exit the program. Below we can see a picture of the front page of the program when it is executed.



```
"E:\L1S3\CSE106 Lab\Project\Project Final\(212002022) CSE-106 Project Code Employee Record Management System.exe"

                  ====LOADING====

 |||||||||||||||||||||||

  46%

 Accessing Main Memory_
```

Figure 3.1.2: Loading Screen Display



```
"E:\L1S3\CSE106 Lab\Project\Project Final\(212002022) CSE-106 Project Code Employee Record Management System.exe"

====== EMPLOYEE RECORD MANAGEMENT SYSTEM ======


           1. Add    Records
           2. View   Records
           3. Modify Records
           4. Search Records
           5. Delete Records
           6. Exit   Program

           Select Your Choice :
```

Figure 3.1.2: Starting Menu Display

3.2 Results and Discussions

Employee Record Management System is a software built to handle the primary housekeeping functions of a company. ERMS helps companies keep track of all the employees and their records. It is used to manage the company using a computerized system. This software built to handle the records of employees of any company. It will help companies to keep track of all the employees' records in a file. The outcomes of the program are as expected. So, I can say, my project is quite successful. There is no system failure in my program. But surely there are more rooms for improvements.

Chapter 4

Conclusion

4.1 Introduction

In a workplace, a huge number of employees may work there. It often becomes a hassle to keep track of the employee's records as hard copies. Managing the hard copies of the records can be time consuming and costly. In order to minimize these types of hassle, Employee Record Management System comes into play. Using it, one can easily manage the employee's records.

4.2 Practical Implications

Basically, it is a simple mid-level program which can be installed in any computer. It is pretty effective and easy to use which will decrease workloads in a company and will increase efficiency in the workplace.

4.3 Scope of Future Work

The project is user friendly and can be updated easily. As it is a cost-effective system, it can easily be commercialized. It can be turned into a full-fledged software for the Govt. sectors and large companies of our country if proper funding is in place.

References

[1] https://aticleworld.com/employee-record-system-project-in-c/

[2] https://www.geeksforgeeks.org/employee-record-system-in-c-using-file-handling/