PROJECT WORK
BY
SUDHA KUMARI SUGASANI

Create Employee Management Application with these requirements
1.Add Employee
Constrains:
Employee Id-Should not be negative(<0)
Should not add existing EmpID
Employee name-Name Should be minimum 3 characters
Employee Salary-Salary must be minimum 10,000
Employee age-Age between 18 and 58
2.a. Search Employee By ID
2.b.Search Employee By Name
3.Display All Employee Details
In Layered Architecture
Employees.txt[Details to be saved in a flat file]

DataAccessLayer Code:

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DataAccessLayer
{
    /// <summary>
    /// This method will return the entered employee details in file
    /// </summary>
    public static class EmployeeDAL
    {
        public static string filePath = "C:\\NH\\Files Using
C#\\Employees.txt";
        public static bool AddEmployee(int empID,String empName,int
empSalary,int empAge)
        {
            try
            {
                string textcontent = string.Concat(empID, ",", empName, ",",
empSalary, ",", empAge);
                File.AppendAllText(filePath, textcontent +
Environment.NewLine);
                return true;
            }
            catch
            {
                return false;
            }
        }
        /// <summary>
        /// This method will check the entered employee id and the existing
employee id,if both are true it will return the matching id employeedetails
        /// </summary>
```

```
        /// <param name="id">int</param>
        /// <returns>employeeFound(employee details if both id's are
matching)</returns>
        public static List<string> GetEmployeesByID(int id)
        {
            var AllEmployees = File.ReadAllLines(filePath);
            bool isfound = false;
            List<string> employeeFound = new List<string>();
            foreach(string employee in AllEmployees)
            {
                var empDetails = employee.Split(',');
                if(Convert.ToInt32(empDetails[0])==id)
                {
                    isfound = true;
                    employeeFound.Add(employee);
                    break;
                }
            }
            return employeeFound;
        }
        /// <summary>
        /// This method will check the entered employeename is present in
existing employee details,if both are true it will return the name which
contains in the employeedetails
        /// </summary>
        /// <param name="name">string</param>
        /// <returns>employeeFound(name that is contained in file)</returns>
        public static List<string> GetEmployeesByName(string name)
        {
            var AllEmployees = File.ReadAllLines(filePath);

            List<string> employeeFound = new List<string>();
            foreach (string employee in AllEmployees)
            {
                var empDetails = employee.Split(',');
                if (empDetails[1].Contains(name))
                {
                    employeeFound.Add(employee);

                }
            }
            return employeeFound;
        }
        /// <summary>
        /// This method will return all employeedetails
        /// </summary>
        /// <returns>allEmployees(data of all employees)</returns>
        public static string[] GetAllEmployees()
        {
            var allEmployees = File.ReadAllLines(filePath);
            return allEmployees;
        }
    }
}
```

Business Logic Library Code:

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
using DataAccessLayer;

namespace BusinessLogicLibrary
{
    public class EmployeeBLL
    {
        /// <summary>
        /// This method will return the entered employee details using DAL if
all the validations are passed
        /// </summary>
        /// <param name="empID">int</param>
        /// <param name="empName">String</param>
        /// <param name="empSalary">int</param>
        /// <param name="empAge">int</param>
        /// <returns>result(var)</returns>
        public static bool AddEmployee(int empID,string empName,int
empSalary,int empAge)
        {
            //string filepath = "C:\\NH\\Files Using C#\\Employees.txt";

             string [] employee = File.ReadAllLines("C:\\NH\\Files Using
C#\\Employees.txt");

            foreach(var e in employee)
            {
                string id = Convert.ToString(empID);

                 if(employee.Equals(id))
                 {
                      Console.WriteLine("Employee id must not be same as
existing employee id");
                       return false;
                 }


            }

            if(empID <= 0&& ((empName.Length) <=3) && empSalary <= 10000 &&
(empAge < 18 || empAge > 58))
            {
                Console.WriteLine("Employee id must be Positive number");
                Console.WriteLine("Employee Name must be atleast 3
characters");
                Console.WriteLine("Employee Salary must be more than 10,000");
                Console.WriteLine("Employee Age must be between 18 and 58");
                return false;

            }
            else if(empID <= 0 && ((empName.Length) < 3) && empSalary <= 10000)
            {
                Console.WriteLine("Employee id must be Positive number");
                Console.WriteLine("Employee Name must be atleast 3
characters");
                Console.WriteLine("Employee Salary must be more than 10,000");
                return false;
            }
            else if(empID <= 0 && ((empName.Length) < 3) && (empAge < 18 ||
empAge > 58))
            {
                Console.WriteLine("Employee id must be Positive number");
                Console.WriteLine("Employee Name must be atleast 3
characters");
                Console.WriteLine("Employee Age must be between 18 and 58");
```

```
                return false;
            }
            else if(((empName.Length) < 3) && empSalary <= 10000 && (empAge <
18 || empAge > 58))
            {
                Console.WriteLine("Employee Name must be atleast 3
characters");
                Console.WriteLine("Employee Salary must be more than 10,000");
                Console.WriteLine("Employee Age must be between 18 and 58");
                return false;
            }
            else if(empID <= 0 && ((empName.Length) < 3))
            {
                Console.WriteLine("Employee id must be Positive number");
                Console.WriteLine("Employee Name must be atleast 3
characters");
                return false;
            }
            else if(empID <= 0 && empSalary <= 10000)
            {
                Console.WriteLine("Employee id must be Positive number");
                Console.WriteLine("Employee Salary must be more than 10,000");
                return false;

            }
            else if(empID <= 0 && (empAge < 18 || empAge > 58))
            {
                Console.WriteLine("Employee id must be Positive number");

                Console.WriteLine("Employee Age must be between 18 and 58");
                return false;
            }
            else if(((empName.Length) < 3) && empSalary < 10000)
            {
                Console.WriteLine("Employee Name must be atleast 3
characters");
                Console.WriteLine("Employee Salary must be more than 10,000");
                return false;
            }
            else if(((empName.Length) < 3)&& (empAge < 18 || empAge > 58))
            {
                Console.WriteLine("Employee Name must be atleast 3
characters");
                Console.WriteLine("Employee Age must be between 18 and 58");
                return false;

            }
            else if(empSalary <= 10000 && (empAge < 18 || empAge > 58))
            {
                Console.WriteLine("Employee Salary must be more than 10,000");

                Console.WriteLine("Employee Age must be between 18 and 58");
                return false;
            }


            else if(empID<=0)
            {
                Console.WriteLine("Employee id must be Positive number");
               return false;
            }
            else if(empName.Length<=3)
            {
```

```
                Console.WriteLine("Employee Name must be atleast 3
characters");
                return false;
            }
            else if(empSalary<=10000)
            {
                Console.WriteLine("Employee Salary must be more than 10,000");
                return false;
            }
            else if(empAge<18||empAge>58)
            {
                Console.WriteLine("Employee Age must be between 18 and 58");
                return false;
            }

                var result = EmployeeDAL.AddEmployee(empID, empName, empSalary,
empAge);
                return result;


        }
        /// <summary>
        /// This method will return the id  using DAL if the entered id is
matching with the existing data
        /// </summary>
        /// <param name="id">int</param>
        /// <returns>result(var)</returns>
        public static List<string> GetEmployeesByID(int id)
        {
            var result = EmployeeDAL.GetEmployeesByID(id);
            return result;
        }
        /// <summary>
        /// This method will return name using DAl if the entered name is
contained in data
        /// </summary>
        /// <param name="name">string</param>
        /// <returns>result(var)</returns>
        public static List<string> GetEmployeesByName(string name)
        {
            var result = EmployeeDAL.GetEmployeesByName(name);
            return result;
        }
        /// <summary>
        /// This method will return all the employee details using DAL
        /// </summary>
        /// <returns>result(var)</returns>
        public static string[] GetAllEmployees()
        {
            var result = EmployeeDAL.GetAllEmployees();
            return result;
        }
    }
}
```

MyClientApp Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using BusinessLogicLibrary;
```

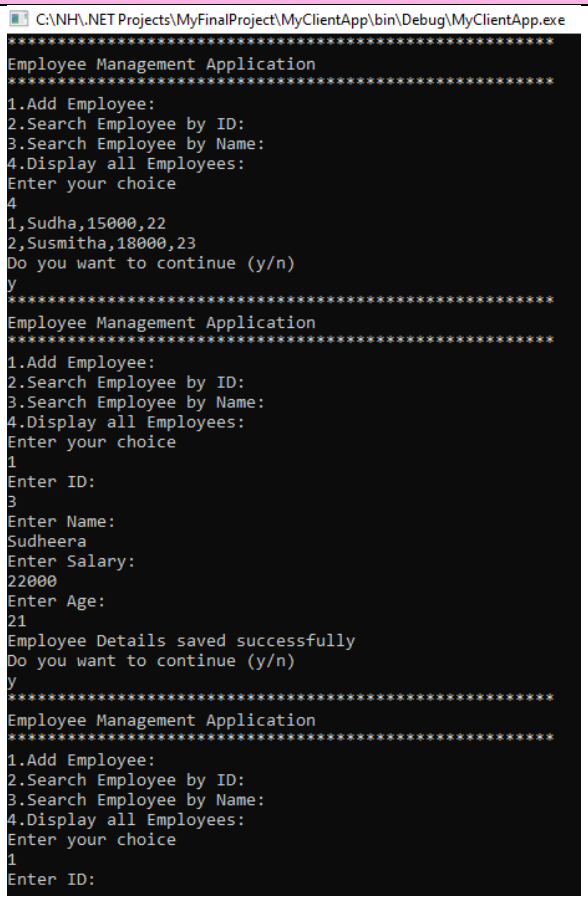
```
namespace MyClientApp
{
    public class Program
    {
        static void Main(string[] args)
        {
            int ch;
            string choice;
            do
            {

Console.WriteLine("*********************************************************");
                Console.WriteLine("Employee Management Application");

Console.WriteLine("*********************************************************");
                Console.WriteLine("1.Add Employee: ");
                Console.WriteLine("2.Search Employee by ID: ");
                Console.WriteLine("3.Search Employee by Name: ");
                Console.WriteLine("4.Display all Employees: ");
                Console.WriteLine("Enter your choice");
                ch = Convert.ToInt32(Console.ReadLine());
                switch (ch)
                {
                    case 1:
                        AddEmployee();
                        break;
                    case 2:
                        SearchEmployeeByID();
                        break;
                    case 3:
                        SearchEmployeeByName();
                        break;
                    case 4:
                        DisplayAllEmployess();
                        break;
                    default:
                        Console.WriteLine("Invalid Option ");
                        break;

                }
                Console.WriteLine("Do you want to continue (y/n) ");
                choice = Console.ReadLine();

            }

            while (choice.Equals("y")) ;
        }
            /// <summary>
            /// This method will take the input from user and store it in the
file using BLL
            /// </summary>
            public static void AddEmployee()
            {
                int id;
                string name;
                int salary;
                int age;
                Console.WriteLine("Enter ID: ");
                id = Convert.ToInt32(Console.ReadLine());
                Console.WriteLine("Enter Name: ");
                name = Console.ReadLine();
                Console.WriteLine("Enter Salary: ");
```

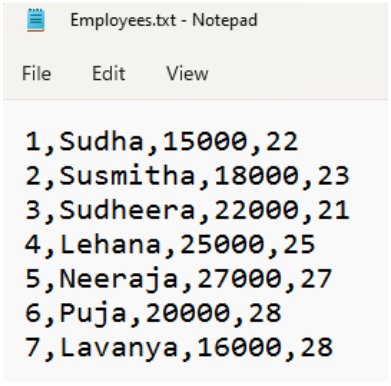
```
                salary = Convert.ToInt32(Console.ReadLine());
                Console.WriteLine("Enter Age: ");
                age = Convert.ToInt32(Console.ReadLine());
                //Calling method in BLL
                var result = EmployeeBLL.AddEmployee(id, name, salary, age);
                if (result)
                {
                    Console.WriteLine("Employee Details saved successfully ");

                }
                else
                {
                    Console.WriteLine("Some error occurred ");
                }

            }
            /// <summary>
            /// This Method will print the EmployeeId if it is found using BLl
            /// </summary>
            public static void SearchEmployeeByID()
            {
                int id;
                Console.WriteLine("Enter id: ");
                id = Convert.ToInt32(Console.ReadLine());
                var result = EmployeeBLL.GetEmployeesByID(id);
                if(result.Count==0)
                {
                    Console.WriteLine("No records found with this id ");
                }
                else
                {
                    result.ForEach(p => Console.WriteLine(p));
                }
            }
            /// <summary>
            /// This method will check the employeename if the given name is
contained in data it'll print names using BLL
            /// </summary>
            public static void SearchEmployeeByName()
            {
                string name;
                Console.WriteLine("Enter name");
                name = Console.ReadLine();
                var result = EmployeeBLL.GetEmployeesByName(name);
                if(result.Count>0)
                {
                    result.ForEach(n => Console.WriteLine(n));
                    //Console.WriteLine(result);
                }
                else
                {
                    Console.WriteLine("No records are matching with this
name");
                }
            }
            /// <summary>
            /// This method will print all Employee details using BLl
            /// </summary>
            public static void DisplayAllEmployess()
            {
                 var result = EmployeeBLL.GetAllEmployees();
                 result.ToList().ForEach(e => Console.WriteLine(e));
               // Console.WriteLine(result);
```

```
            }
        }
}
```

Output:

Entering correct details and stored in flat file:





Search by EmployeeID:

```
Do you want to continue (y/n)
y
****************************************
Employee Management Application
****************************************
1.Add Employee:
2.Search Employee by ID:
3.Search Employee by Name:
4.Display all Employees:
Enter your choice
2
Enter id:
4
4,Lehana,25000,25
Do you want to continue (y/n)
```

Search by EmployeeName:

```
Do you want to continue (y/n)
y
****************************************
Employee Management Application
****************************************
1.Add Employee:
2.Search Employee by ID:
3.Search Employee by Name:
4.Display all Employees:
Enter your choice
3
Enter name
Su
1,Sudha,15000,22
2,Susmitha,18000,23
3,Sudheera,22000,21
Do you want to continue (y/n)
```

Display All Employee Details:

```
****************************************
Employee Management Application
****************************************
1.Add Employee:
2.Search Employee by ID:
3.Search Employee by Name:
4.Display all Employees:
Enter your choice
4
1,Sudha,15000,22
2,Susmitha,18000,23
3,Sudheera,22000,21
4,Lehana,25000,25
5,Neeraja,27000,27
6,Puja,20000,28
7,Lavanya,16000,28
Do you want to continue (y/n)
```

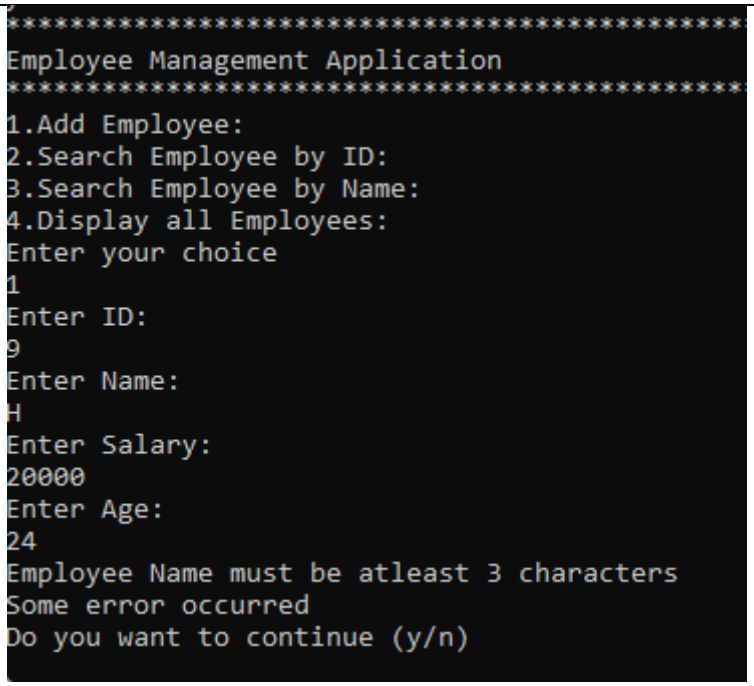
Validations:

C:\NH\.NET Projects\MyFinalProject\MyClientApp\bin\Debug\My

```
****************************************************
Employee Management Application
****************************************************
1.Add Employee:
2.Search Employee by ID:
3.Search Employee by Name:
4.Display all Employees:
Enter your choice
1
Enter ID:
0
Enter Name:
Si
Enter Salary:
2000
Enter Age:
12
Employee id must be Positive number
Employee Name must be atleast 3 characters
Employee Salary must be more than 10,000
Employee Age must be between 18 and 58
Some error occurred
Do you want to continue (y/n)
```

```
*********************************************
Employee Management Application
*********************************************
1.Add Employee:
2.Search Employee by ID:
3.Search Employee by Name:
4.Display all Employees:
Enter your choice
1
Enter ID:
8
Enter Name:
De
Enter Salary:
1500
Enter Age:
10
Employee Name must be atleast 3 characters
Employee Salary must be more than 10,000
Employee Age must be between 18 and 58
Some error occurred
Do you want to continue (y/n)
```

```
C:\NH\.NET Projects\MyFinalProject\MyClientApp\bin\Deb
*********************************************
Employee Management Application
*********************************************
1.Add Employee:
2.Search Employee by ID:
3.Search Employee by Name:
4.Display all Employees:
Enter your choice
1
Enter ID:
9
Enter Name:
Hari
Enter Salary:
2500
Enter Age:
78
Employee Salary must be more than 10,000
Employee Age must be between 18 and 58
Some error occurred
Do you want to continue (y/n)
```


****
Employee Management Application
****
1.Add Employee:
2.Search Employee by ID:
3.Search Employee by Name:
4.Display all Employees:
Enter your choice
1
Enter ID:
9
Enter Name:
H
Enter Salary:
20000
Enter Age:
24
Employee Name must be atleast 3 characters
Some error occurred
Do you want to continue (y/n)