|  |
| --- |
| **Final Project**  **By**  **Nanam VAishnavi**  **22 – feb - 2022** |

|  |
| --- |
| **Employee Management Application**   * **Add Employee** * **Search Employee By ID** * **Search Employee By Name** * **Display All Employees**   **======================================================**   1. **Employee ID (Should not be negative and not repeated)** 2. **Employee Name (minimum 3 Characters)** 3. **Employee Salary (minimum 10000)** 4. **Employee Age (age>=18 and age<=58)** |

|  |
| --- |
| **MyClientApp** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using BusinessLogicLayer;  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  // Author : Nanam Vaishnavi  // Purpose : MyClientApp Console Application  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace MyClientApp  {  internal 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 Employee: ");  Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  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:  DisplayAllEmployees();  break;  default:  Console.WriteLine("Invalid Option");  break;  }  Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  Console.WriteLine("Do you want to continue(y/n)");  choice = Console.ReadLine();  } while (choice == "y");  }  public static void AddEmployee()  {  int id, salary, age;  string name;  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());  var result = EmployeeBLL.AddEmployee(id, name, salary, age);  Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  if (result)  Console.WriteLine("Employee Details Saved Successfully");  else  Console.WriteLine("Error Occured");  }  public static void SearchEmployeeById()  {    int id;  Console.WriteLine("Enter ID: ");  id = Convert.ToInt32(Console.ReadLine());  var result = EmployeeBLL.GetEmployeeById(id);  Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  if (result.Count == 0)  Console.WriteLine("No records exists with this ID ");  else    result.ForEach(r => Console.WriteLine(r));  }  public static void SearchEmployeeByName()  {  string name;  Console.WriteLine("Enter Name: ");  name = Console.ReadLine();  var result = EmployeeBLL.GetEmployeeByName(name);  if (result.Count == 1)  result.ForEach(r => Console.WriteLine(r));    else  {  Console.WriteLine("No records exists with this NAME ");  }  }  public static void DisplayAllEmployees()  {  var result = EmployeeBLL.GetAllEmployees();  foreach (var employee in result)  {  Console.WriteLine(employee);  }  }  }  } |
| **Business Logic Layer Code** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using DataAccessLayer;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  // Author : Nanam Vaishnavi  // Purpose : Business Logic Layer Class Library  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace BusinessLogicLayer  {  public class EmployeeBLL  {  public static bool AddEmployee(int empId, string empName, int empSalary, int empAge)  {  var result = EmployeeDAL.AddEmployee(empId, empName, empSalary, empAge);    return result;  }  public static List<String> GetEmployeeById(int id)  {  var result = EmployeeDAL.GetEmployeeById(id);  return result;  }  public static List<String> GetEmployeeByName(string name)  {  var result = EmployeeDAL.GetEmployeeByName(name);    return result;  }  public static string[] GetAllEmployees()  {  var result = EmployeeDAL.GetAllEmployees();  return result;  }  }  } |
| **Data Access Layer Code** |
| using System;  using System.Collections.Generic;  using System.IO;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  // Author : Nanam Vaishnavi  // Purpose : Data Access Layer Class Library  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace DataAccessLayer  {  public class EmployeeDAL  {  /// <summary>  /// Filepath  /// </summary>  public static string filepath = "F:\\NH\\EmployeesData\\Employee.txt";  /// <summary>  /// Add Employee  /// </summary>  /// <param name="empId"></param>  /// <param name="empName"></param>  /// <param name="empSalary"></param>  /// <param name="empAge"></param>  /// <returns></returns>  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 (Exception ex)  {  return false;  }  }  /// <summary>  /// GetEmployeeById()  /// </summary>  /// <param name="id"></param>  /// <returns></returns>  public static List<String> GetEmployeeById(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>  /// GetEmployeeByName  /// </summary>  /// <param name="name"></param>  /// <returns></returns>  public static List<String> GetEmployeeByName(string name)  {  var allEmployees = File.ReadAllLines(filepath);  bool isFound = false;  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>  /// GetAllEmployees  /// </summary>  /// <returns></returns>  public static string[] GetAllEmployees()  {  var allEmployees = File.ReadAllLines(filepath);  return allEmployees;  }  }  } |

|  |
| --- |
| **Solution Explorer** |
| 1. **Add Employee** |

|  |
| --- |
| 1. **Search Employee By ID**      1. **Search Employee By Name**      1. **Display All Employees** |