2/22/2022

DAY22 ASSIGNMENT

EMPLOYEE MANAGEMENT APPLICATION

Bhanu Prakash Reddy Chilukuri

NB HEALTHCARE TECHNOLOGIES

Employee Management Application

Add Employee

Search Employee

Display All Employees

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

1.Employee ID (Should not be negative)

2.Employee Name (min 3 Characters)

3.Employee Salary (min 10000)

4.Employee Age (age >=18 and age <=58)

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Employees.txt [to be saved in file]

Application Code:

Data Access Layer:

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DataAccessLibraray

{

//-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-

//Author: Bhanu Prakash Reddy

//WACP for Employee Data Access Layer

//-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-

public static class EmployeesDAL

{

/// <summary>

/// filepath for saved text file

/// </summary>

public static string filepath = "D:\\Employee Data\\Employee.txt";

/// <summary>

/// Adding Employee Details

/// </summary>

/// <param name="empID"></param>

/// <param name="empName"></param>

/// <param name="empSalary"></param>

/// <param name="empAge"></param>

/// <returns>Employee Data</returns>

public static bool AddEmployee(int empID, string empName, int empSalary, int empAge)

{

try

{

string empdetails = string.Concat(empID, ",", empName, ",", empSalary, ",", empAge);

File.AppendAllText(filepath, empdetails + Environment.NewLine);

return true;

}

catch

{

return false;

}

}

/// <summary>

/// Finding empolyee data by using Id

/// </summary>

/// <param name="ID"></param>

/// <returns>eEmployee Data</returns>

public static List<string> FindEmployeesByID(int ID)

{

var allEmployees = File.ReadAllLines(filepath);

bool isFound = false;

List<string> employeesFound = new List<string>();

foreach (string employee in allEmployees)

{

var empdetails = employee.Split(',');

if (Convert.ToInt32(empdetails[0]) == ID)

{

isFound = true;

employeesFound.Add(employee);

break;

}

}

return employeesFound;

}

/// <summary>

/// Finding employee data by using name

/// </summary>

/// <param name="name"></param>

/// <returns>Employee data</returns>

public static List<string> FindEmployeesByName(string name)

{

var allEmployees = File.ReadAllLines(filepath);

List<string> employeesFound = new List<string>();

foreach (string employee in allEmployees)

{

var empdetails = employee.Split(',');

if (empdetails[1].Contains(name))

{

employeesFound.Add(employee);

}

}

return employeesFound;

}

/// <summary>

/// Dispalying All Employees Data

/// </summary>

/// <returns>All Employee Data</returns>

public static string[] FindAllEmployees()

{

var allEmployees = File.ReadAllLines(filepath);

return allEmployees;

}

}

}

Business Logic Layer:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using DataAccessLibraray;

namespace BusinessLogicLibraray

{

//-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-

//Author: Bhanu Prakash Reddy

//WACP for Employee Business Logic Layer

//-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-

public class EmployeesBLL

{

public static bool AddEmployee(int empID, string empName, int empSalary, int empAge)

{

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

return result;

}

public static List<string> FindEmployeesByID(int ID)

{

var result = EmployeesDAL.FindEmployeesByID(ID);

return result;

}

public static List<string> FindEmployeesByName(string name)

{

var result = EmployeesDAL.FindEmployeesByName(name);

return result;

}

public static string[] FindAllEmployees()

{

var result = EmployeesDAL.FindAllEmployees();

return result;

}

}

}

Client App:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using BusinessLogicLibraray;

namespace BhanuClientApp

{

public class Program

{

static void Main(string[] args)

{

int s;

string choice;

do

{

Console.WriteLine("-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-");

Console.WriteLine("Employee Management Application");

Console.WriteLine("-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-=x=-");

Console.WriteLine("1. Add Employee : ");

Console.WriteLine("2. Search Employee Details by Id : ");

Console.WriteLine("3. Search Employee Details by Name : ");

Console.WriteLine("4. Display All Employee Details : ");

Console.WriteLine("Enter your Choice");

s = Convert.ToInt32(Console.ReadLine());

switch (s)

{

case 1:

AddEmployee();

break;

case 2:

FindEmployeesByID();

break;

case 3:

FindEmployeesByName();

break;

case 4:

FindAllEmployees();

break;

default:

Console.WriteLine("Enter valid option");

break;

}

Console.WriteLine("Do you want to continue(y/n):");

choice = Console.ReadLine();

}

while (choice.Equals("y"));

}

public static void AddEmployee()

{

int id, salary, age;

string name;

Console.WriteLine("Enter Employee Id: ");

id= Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter Employee Name: ");

name = Console.ReadLine();

Console.WriteLine("Enter Employee Salary: ");

salary = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter Employee Age: ");

age = Convert.ToInt32(Console.ReadLine());

var empData = EmployeesBLL.AddEmployee(id, name, salary, age);

if(empData)

Console.WriteLine("Data Added Sucessfully");

else

Console.WriteLine("Enter valid data");

}

public static void FindEmployeesByID()

{

int id;

Console.WriteLine("Enter Employee Id : ");

id = Convert.ToInt32(Console.ReadLine());

var empData = EmployeesBLL.FindEmployeesByID(id);

if(empData.Count==0)

Console.WriteLine($"No data exists in this {id}");

else

{

empData.ForEach(emp => Console.WriteLine(emp));

}

}

public static void FindEmployeesByName()

{

string name;

Console.WriteLine("Enter Employee Name : ");

name = Console.ReadLine();

var empData = EmployeesBLL.FindEmployeesByName(name);

if (empData.Count == 0)

Console.WriteLine($"No data exists in this {name}");

else

{

empData.ForEach(emp => Console.WriteLine(emp));

}

}

public static void FindAllEmployees()

{

var empData = EmployeesBLL.FindAllEmployees();

foreach(var emp in empData)

Console.WriteLine(emp);

}

}

}

Output:









