Program

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace EmployeeProjectSolution

{

class Program

{

static void Main(string[] args)

{

Program program = new Program();

program.manageMenu();

Console.ReadKey();

}

void manageMenu()

{

ManageMenu mm = new ManageMenu();

Console.WriteLine("Welcome to Employee Menu Management");

int choice = 0;

do

{

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

Console.WriteLine("2: Edit Employee Age : ");

Console.WriteLine("3: Print Employee Details : ");

Console.WriteLine("4: Delete an Employee : ");

Console.WriteLine("5: Exit");

while (!int.TryParse(Console.ReadLine(), out choice))

{

Console.WriteLine("Please enter a number");

}

switch (choice)

{

case 1:

mm.AddEmployee();

break;

case 2:

mm.EditEmployeeAge();

break;

case 3:

mm.GetAllEmployee();

break;

case 4:

mm.DeleteEmployee();

break;

case 0:

Console.WriteLine("Exit");

break;

default:

Console.WriteLine("Invalid choice. Please try again");

break;

}

}

while (choice != 0);

}

}

}

Manage Menu

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using EmployeeDALLibrary;

using EmployeeModelsLibrary;

namespace EmployeeProjectSolution

{

internal class ManageMenu

{

List <Employee> employees;

EmployeeDAL employeeDAL;

public ManageMenu()

{

employeeDAL = new EmployeeDAL();

}

public void GetAllEmployee()

{

employees = null;

try

{

var employees = employeeDAL.GetAllEmployees().ToList();

}

catch (NoRecordException nre)

{

Console.WriteLine(nre.Message);

}

catch (Exception npe)

{

Console.WriteLine("Something went wrong. Will fix soon...");

Console.WriteLine(npe.Message);

}

}

public void AddEmployee()

{

Employee employee = new Employee();

employee.GetEmployeeDetails();

try

{

employeeDAL.InsertNewEmployee(employee);

}

catch (Exception e)

{

Console.WriteLine("Could not add the employee");

Console.WriteLine(e.Message);

}

}

public void EditEmployeeAge()

{

int id = GetIdFromUser();

Employee employee = GetEmployeeById(id);

if (employee == null)

{

Console.WriteLine("Invalid Id. Cannot edit");

return;

}

try

{

employee.GetEmployeeDetails();

employeeDAL.UpdateEmployeeAge(employee);

}

catch (Exception e)

{

Console.WriteLine("Could not update the employee");

Console.WriteLine(e.Message);

}

Console.WriteLine("Update employee as below: ");

PrintEmployee(employee);

}

public void DeleteEmployee()

{

int id = GetIdFromUser();

Employee employee = GetEmployeeById(id);

if (employee == null)

{

Console.WriteLine("Invalid Id. Cannot edit");

return;

}

try

{

employeeDAL.DeleteEmployee(id);

}

catch (Exception e)

{

Console.WriteLine("Could not remove the employee");

Console.WriteLine(e.Message);

}

Console.WriteLine("Update employee as below: ");

PrintEmployee(employee);

}

public Employee GetEmployeeById(int id)

{

GetAllEmployee();

Employee employee = employees.SingleOrDefault(p => p.Id == id);

return employee;

}

int GetIdFromUser()

{

Console.WriteLine("Please enter the employee id");

int id;

while (!int.TryParse(Console.ReadLine(), out id))

{

Console.WriteLine("Invalid entry ID. Please try again...");

}

return id;

}

public void PrintEmployees()

{

GetAllEmployee();

var sortedEmployees = employees.OrderBy(p => p.Id);

foreach (var item in sortedEmployees)

{

if (item != null)

PrintEmployee(item);

}

}

public void PrintEmployeeById()

{

int id = GetIdFromUser();

Employee employee = GetEmployeeById(id);

if (employee != null)

{

PrintEmployee(employee);

}

else

Console.WriteLine("No such employee");

}

private void PrintEmployee(Employee item)

{

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine(item);

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

}

}

EmployeeDAL

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using EmployeeModelsLibrary;

using System.Data.SqlClient;

using System.Data;

namespace EmployeeDALLibrary

{

public class EmployeeDAL

{

SqlConnection conn;

public EmployeeDAL()

{

conn = myConnection.GetConnection();

}

public ICollection<Employee> GetAllEmployees()

{

if (conn.State == ConnectionState.Open)

conn.Close();

List<Employee> employees = new List<Employee>();

DataSet ds = new DataSet();

SqlDataAdapter adapter = new SqlDataAdapter("proc\_GetAllEmployee", conn);

adapter.SelectCommand.CommandType = CommandType.StoredProcedure;

adapter.Fill(ds);

Employee employee;

if (ds.Tables[0].Rows.Count != 0)

throw new NoRecordException();

foreach (DataRow item in ds.Tables[0].Rows)

{

employee = new Employee();

employee.Id = Convert.ToInt32(item[0]);

employee.Name = item[1].ToString();

employee.Age = Convert.ToInt32(item[2]);

employees.Add(employee);

}

return employees;

}

public bool InsertNewEmployee(Employee employee)

{

if (conn.State == ConnectionState.Open)

conn.Close();

SqlCommand cmd = new SqlCommand("proc\_AddEmployee", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@emp\_name", employee.Name);

cmd.Parameters.AddWithValue("@emp\_age", employee.Age);

conn.Open();

if (cmd.ExecuteNonQuery() > 0)

{

return true;

}

return false;

}

public bool UpdateEmployeeAge(Employee employee)

{

if (conn.State == ConnectionState.Open)

conn.Close();

SqlCommand cmd = new SqlCommand("proc\_EditEmployeeAge", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@emp\_id", employee.Id);

cmd.Parameters.AddWithValue("@emp\_age", employee.Age);

conn.Open();

if (cmd.ExecuteNonQuery() > 0)

{

return true;

}

return false;

}

public bool DeleteEmployee(int id)

{

if (conn.State == ConnectionState.Open)

conn.Close();

SqlCommand cmd = new SqlCommand("proc\_DeleteEmployee", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@emp\_id", id);

conn.Open();

if (cmd.ExecuteNonQuery() > 0)

{

return true;

}

return false;

}

//public Employee GetEmployeeByID(int id)

//{

// var employee = GetAllEmployees().SingleOrDefault(x => x.Id == id);

// if (employee != null)

// {

// return employee;

// }

// return null;

//}

}

}

Employee.CS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace EmployeeModelsLibrary

{

public class Employee

{

public int Id { get; set; }

public string Name { get; set; }

public int Age { get; set; }

public void GetEmployeeDetails()

{

Console.WriteLine("Please enter the employee name :");

Name = Console.ReadLine();

Console.WriteLine("Please enter the employee's age :");

int age;

while (!int.TryParse(Console.ReadLine(), out age))

{

Console.WriteLine("Invalid Age. Please enter again: ");

}

Age = age;

}

public override string ToString()

{

return "Employee ID " + Id

+ "\nEmployee Name " + Name

+ "\nEmployee Age" + Age;

}

}

}