public class Department

{

[Key]

public int Department\_Id { get; set; }

public string Name { get; set; }

public ICollection<Employee> Employees { get; set; }

public void GetNewDepartmentDetails()

{

Console.WriteLine("Please enter department name");

Name = Console.ReadLine();

}

public override string ToString()

{

return "Department ID " + Department\_Id

+ "\nName " + Name;

}

}

public class Employee

{

[Key]

public int Id { get; set; }

public string Name { get; set; }

public int Age { get; set; }

public int Department\_Id { get; set; }

[ForeignKey("Department\_Id")]

public Department Department { get; set; }

public void GetNewEmployeeDetails()

{

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

Name = Console.ReadLine();

Console.WriteLine("Please enter employee age");

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

Console.WriteLine("Please enter department id");

Department\_Id = Convert.ToInt32(Console.ReadLine());

}

public override string ToString()

{

return "Employee ID " + Id

+ "\nName " + Name

+ "\nAge "+Age

+ "\nDepartment Id "+Department\_Id;

}

}

internal class CompanyContext : DbContext

{

public CompanyContext() : base("conn")

{

}

public DbSet<Employee> Employees { get; set; }

public DbSet<Department> Departments { get; set; }

}

class Program

{

void manageMenu()

{

int choice = 0;

do

{

ManageMenu manageMenu = new ManageMenu();

Console.WriteLine("Welcome");

Console.WriteLine("1. Add department");

Console.WriteLine("2. Edit department name");

Console.WriteLine("3. Print Departments");

Console.WriteLine("4. Add employee");

Console.WriteLine("5. Edit employee age");

Console.WriteLine("6. Edit employee department");

Console.WriteLine("7. Print all employees");

Console.WriteLine("8. Exit");

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

{

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

}

try

{

switch (choice)

{

case 1:

manageMenu.AddDepartment();

break;

case 2:

manageMenu.EditDepartmentName();

break;

case 3:

manageMenu.GetAllDepartments();

break;

case 4:

manageMenu.AddEmployee();

break;

case 5:

manageMenu.EditEmployeeAge();

break;

case 6:

manageMenu.EditEmployeeDepartment();

break;

case 7:

manageMenu.GetAllEmployees();

break;

case 8:

Console.WriteLine("bye");

break;

default:

Console.WriteLine("Invalid choice, pls try again");

break;

}

}

catch (NullReferenceException nre)

{

Console.WriteLine("null mistake");

Console.WriteLine(nre.Message);

}

catch (ArgumentOutOfRangeException aore)

{

Console.WriteLine("Pizza could not be found");

Console.WriteLine(aore.Message);

}

catch (FormatException fe)

{

Console.WriteLine("expecting a number");

Console.WriteLine(fe.Message);

}

catch (Exception e)

{

Console.WriteLine("oops something went wrong");

Console.WriteLine(e.Message);

}

} while (choice != 8);

}

static void Main(string[] args)

{

Program program = new Program();

program.manageMenu();

Console.ReadKey();

}

}

internal class ManageMenu

{

CompanyDAL companyDAL;

ICollection<Department> departments;

ICollection<Employee> employees;

public ManageMenu()

{

companyDAL = new CompanyDAL();

}

public void GetAllDepartments()

{

departments = null;

try

{

departments = companyDAL.GetAllDepartments();

}

catch (Exception e)

{

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

Console.WriteLine(e.Message);

}

PrintDepartments();

}

private void PrintDepartment(Department department)

{

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

Console.WriteLine(department);

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

}

public void PrintDepartments()

{

var sortedDepartments = departments.OrderBy(p => p.Department\_Id);

foreach (var item in sortedDepartments)

{

if (item != null)

//Console.WriteLine(item);

PrintDepartment(item);

}

}

public void GetAllEmployees()

{

employees = null;

try

{

employees = companyDAL.GetAllEmployees();

}

catch (Exception e)

{

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

Console.WriteLine(e.Message);

}

PrintEmployees();

}

private void PrintEmployee(Employee employee)

{

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

Console.WriteLine(employee);

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

}

public void PrintEmployees()

{

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

foreach (var item in sortedEmployees)

{

if (item != null)

//Console.WriteLine(item);

PrintEmployee(item);

}

}

public void AddEmployee()

{

Employee employee = new Employee();

employee.GetNewEmployeeDetails();

try

{

companyDAL.InsertNewEmployee(employee);

}

catch (Exception e)

{

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

Console.WriteLine(e.Message);

}

}

public void AddDepartment()

{

Department department = new Department();

department.GetNewDepartmentDetails();

try

{

companyDAL.InsertNewDepartment(department);

}

catch (Exception e)

{

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

Console.WriteLine(e.Message);

}

}

public void EditDepartmentName()

{

GetAllDepartments();

int id = GetDepartmentIdFromUser();

string name = GetDepartmentNameToEditFromUser();

try

{

companyDAL.UpdateDepartmentName(id, name);

}

catch (Exception e)

{

Console.WriteLine("Could not change the department name");

Console.WriteLine(e.Message);

}

}

public void EditEmployeeAge()

{

GetAllEmployees();

int id = GetEmployeeIdFromUser();

int age = GetAgeToEditFromEmployee();

try

{

companyDAL.UpdateEmployeeAge(id, age);

}

catch (Exception e)

{

Console.WriteLine("Could not change employee age");

Console.WriteLine(e.Message);

}

}

int GetAgeToEditFromEmployee()

{

Console.WriteLine("Please enter the age you want to change to");

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

return age;

}

string GetDepartmentNameToEditFromUser()

{

Console.WriteLine("Please enter the department name you want to change to");

string name = Console.ReadLine();

return name;

}

int GetDepartmentIdFromUser()

{

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

int id;

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

{

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

}

return id;

}

int GetEmployeeIdFromUser()

{

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

int id;

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

{

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

}

return id;

}

public void EditEmployeeDepartment()

{

GetAllEmployees();

int id = GetEmployeeIdFromUser();

int departmentId = GetEmployeeDepartmentId();

try

{

companyDAL.UpdateEmployeeDepartment(id, departmentId);

}

catch (Exception e)

{

Console.WriteLine("Could not change employee department");

Console.WriteLine(e.Message);

}

}

int GetEmployeeDepartmentId()

{

GetAllDepartments();

Console.WriteLine("Please enter your new department id");

int id;

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

{

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

}

return id;

}

}

public class CompanyDAL

{

readonly CompanyContext \_companyContext;

public CompanyDAL()

{

\_companyContext = new CompanyContext();

}

public ICollection<Department> GetAllDepartments()

{

List<Department> departments = \_companyContext.Departments.ToList();

if (departments.Count == 0)

Console.WriteLine("No departments available");

return departments;

}

public ICollection<Employee> GetAllEmployees()

{

List<Employee> employees = \_companyContext.Employees.ToList();

if (employees.Count == 0)

Console.WriteLine("No employees available");

return employees;

}

public void InsertNewDepartment(Department department)

{

\_companyContext.Departments.Add(department);

\_companyContext.SaveChanges();

Console.WriteLine("Department added");

}

public void InsertNewEmployee(Employee employee)

{

\_companyContext.Employees.Add(employee);

\_companyContext.SaveChanges();

Console.WriteLine("Employee added");

}

public void UpdateDepartmentName(int id, string name)

{

Department department = \_companyContext.Departments.SingleOrDefault(p => p.Department\_Id == id);

if (department == null)

{

Console.WriteLine("no such department");

return;

}

department.Name = name;

\_companyContext.SaveChanges();

Console.WriteLine("Department name editted");

}

public void UpdateEmployeeAge(int id, int age)

{

Employee employee = \_companyContext.Employees.SingleOrDefault(p => p.Id == id);

if (employee == null)

{

Console.WriteLine("no such employee");

return;

}

employee.Age = age;

\_companyContext.SaveChanges();

Console.WriteLine("Employee age editted");

}

public void UpdateEmployeeDepartment(int id, int deptId)

{

Employee employee = \_companyContext.Employees.SingleOrDefault(p => p.Id == id);

if (employee == null)

{

Console.WriteLine("no such employee");

return;

}

employee.Department\_Id = deptId;

\_companyContext.SaveChanges();

Console.WriteLine("Employee department editted");

}

}







