

Employee Model

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

namespace Day2Task.Model
{
    public class EmployeeModel
    {
        public int Id { get; set; }

        public string Name { get; set; }
        public string Designation { get; set; }

        public DateTime DOJ { get; set; }

        public EmployeeModel(int id,string name,string designation, DateTime doj) {
            this.Id=id;
            this.Name=name;
            this.Designation=designation;
            this.DOJ=doj;
        }
    }
}
```

ManagerModel

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

namespace Day2Task.Model
{
    public class ManagerModel : EmployeeModel
    {
        public int NoofEmployees {get; set;}
        public ManagerModel(int Id,string Name,string Designation,DateTime Date,int
NoofEmployees):
            base(Id,Name,Designation,Date) {
            this.NoofEmployees = NoofEmployees;
        }
    }
}
```

IEmployeeRepo

```
using Day2Task.Model;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day2Task.Repo
{
    public interface IEmployeeRepo
    {
        public void AddEmployee(EmployeeModel Employee);

        public void GetAllEmployees();

        public void GetEmployeeById(int Id);
    }
}
```

IManagerRepo

```
using Day2Task.Model;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day2Task.Repo
{
    public interface IManagerRepo
    {

        public void GetAllManagers();
        public void GetManagerByID(int Id);

        public void AddEmployeestoManager(ManagerModel Manager );
        public void UpdateEmployeestoManager(int Id, int newEmployeeCount);

    }
}
```

EmployeeRepo

```
using Day2Task.Model;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day2Task.Repo
{
    public class EmployeeRepo : IEmployeeRepo
    {
        List<EmployeeModel> EmployeesList = new List<EmployeeModel>();
        public void GetAllEmployees()
        {
            foreach (var employee in EmployeesList)
            {
                Console.WriteLine($"ID : {employee.Id} Name : {employee.Name}");
            }
        }
    }
}
```

```
    }  
}  
  
public void AddEmployee(EmployeeModel Employee)  
{  
    EmployeesList.Add(Employee);  
  
    foreach (var emp in EmployeesList)  
    {  
        Console.WriteLine($"ID: {emp.Id}, Name: {emp.Name}");  
    }  
}  
  
public void GetEmployeeById(int Id)  
{  
    var employee = EmployeesList.FirstOrDefault(emp => emp.Id == Id);  
    if (employee != null)  
    {  
        Console.WriteLine($"Employee found: {employee.Name}, ID: {employee.Id}");  
    }  
    else  
    {  
        Console.WriteLine("Employee not found");  
    }  
}  
}  
}
```

ManagerRepo

```
using Day2Task.Model;  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace Day2Task.Repo  
{  
    public class ManagerRepo : IManagerRepo  
    {
```

```
List<ManagerModel> ManagersList = new List<ManagerModel>();
public void GetAllManagers()
{
    foreach (var manager in ManagersList)
    {
        Console.WriteLine($"ID: {manager.Id}, Name: {manager.Name}, No of Employees:
{manager.NoofEmployees}");
    }
}

public void GetManagerByID(int Id)
{
    var manager = ManagersList.FirstOrDefault(mgr => mgr.Id == Id);
    if (manager != null)
    {
        Console.WriteLine($"Manager found: {manager.Name}, ID: {manager.Id}, No of
Employees: {manager.NoofEmployees}");
    }
    else
    {
        Console.WriteLine("Manager not found");
    }
}

public void UpdateEmployeestoManager(int Id, int newEmployeeCount)
{
    var manager = ManagersList.FirstOrDefault(mgr => mgr.Id == Id);
    if (manager != null)
    {
        manager.NoofEmployees = newEmployeeCount;
        Console.WriteLine($"Manager {manager.Name} now manages
{manager.NoofEmployees} employees.");
    }
    else
    {
        Console.WriteLine("Manager not found");
    }
}

public void AddEmployeestoManager(ManagerModel Manager)
{
    ManagersList.Add(Manager);
    foreach (var mgr in ManagersList)
    {
```

```
        Console.WriteLine($"ID: {mgr.Id}, Name: {mgr.Name}, No of Employees:  
{mgr.NoofEmployees}");  
    }  
}
```

```
}  
}
```

OUTPUT

```
Choose an option:
1. Employee
2. Manager
3. Exit
1

Employee Operations:
1. Add Employee
2. Get All Employees
3. Get Employee by Id
4. Back
1
Enter Id: 7
Enter Name: Karthi
Enter Designation: IT
Enter Date (yyyy-MM-dd): 2024-10-10
ID: 7, Name: Karthi

Employee Operations:
1. Add Employee
2. Get All Employees
3. Get Employee by Id
4. Back
2
ID : 7 Name : Karthi

Employee Operations:
1. Add Employee
2. Get All Employees
3. Get Employee by Id
4. Back
3
Enter Employee Id: 7
Employee found: Karthi, ID: 7

Employee Operations:
1. Add Employee
2. Get All Employees
3. Get Employee by Id
4. Back
```

Choose an option:

1. Employee
2. Manager
3. Exit

2

Manager Operations:

1. Add Manager
2. Get All Managers
3. Get Manager by Id
4. Update Employees count for Manager
5. Back

1

Enter Id: 8

Enter Name: Surya

Enter Designation: SA

Enter Date (yyyy-MM-dd): 2020-10-10

Enter Number of Employees managed: 30

ID: 8, Name: Surya, No of Employees: 30

Manager Operations:

1. Add Manager
2. Get All Managers
3. Get Manager by Id
4. Update Employees count for Manager
5. Back

2

ID: 8, Name: Surya, No of Employees: 30

Manager Operations:

1. Add Manager
2. Get All Managers
3. Get Manager by Id
4. Update Employees count for Manager
5. Back

3

Enter Manager Id: 8

Manager found: Surya, ID: 8, No of Employees: 30


```
Manager Operations:
1. Add Manager
2. Get All Managers
3. Get Manager by Id
4. Update Employees count for Manager
5. Back
4
Enter Manager Id: 8
Enter new Employee count: 35
Manager Surya now manages 35 employees.
```

```
Manager Operations:
1. Add Manager
2. Get All Managers
3. Get Manager by Id
4. Update Employees count for Manager
5. Back
2
ID: 8, Name: Surya, No of Employees: 35
```

```
Manager Operations:
1. Add Manager
2. Get All Managers
3. Get Manager by Id
4. Update Employees count for Manager
5. Back
```