1. Player.cs :

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

namespace PhaseEndProject
{
   public class Player
   {
      public int PlayerId { get; set; }

      public string PlayerName { get; set; }

      public int PlayerAge { get; set; }

   }
}
```

2. ITeam.cs :

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

namespace PhaseEndProject
{
   public  interface ITeam
   {
      void Add(Player player);

      void Remove(int playerId);

      List<Player> GetAllPlayers();

      Player GetPlayerById(int playerId);

      Player GetPlayerByName(string playerName);

   }
}
```

3. OneDayTeam.cs :

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

```
namespace PhaseEndProject
{
    public class OneDayTeam:ITeam
    {
        public static List<Player> oneDayTeam = new List<Player>();

        public OneDayTeam()
        {
            oneDayTeam.Capacity = 11;
        }
        public void Add(Player player)
        {
            if (oneDayTeam.Count < 11)
            {
                oneDayTeam.Add(player);
            }
        }

        public void Remove(int playerId)
        {
            var player = oneDayTeam.Where(p => p.PlayerId == playerId).FirstOrDefault();
            oneDayTeam.Remove(player);
        }

        public List<Player> GetAllPlayers()
        {
           return oneDayTeam.ToList();
        }

        public Player GetPlayerById(int playerId)
        {
            var player = oneDayTeam.Where(p=>p.PlayerId==playerId).FirstOrDefault();
            return player;
        }

        public Player GetPlayerByName(string playerName)
        {
            var player = oneDayTeam.Where(p => p.PlayerName == playerName).FirstOrDefault();
            return player;
        }
    }
}
```

4. Program.cs :

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

namespace PhaseEndProject
{
    internal class Program
    {
        static void Main()
        {
            OneDayTeam team1 = new OneDayTeam();
        start: Console.Write("Enter 1:To Add Player 2:To Remove Player by Id 3.Get Player By Id 4.Get Player by Na
me 5.Get All Players:");
            int choice = int.Parse(Console.ReadLine());

            switch (choice)
            {
                case 1:
                    Console.Write("Enter Player Id:");
                    int id = int.Parse(Console.ReadLine());
                    Console.Write("Enter Player Name:");
                    string name = Console.ReadLine();
                    Console.Write("Enter Player Age:");
                    int age = int.Parse(Console.ReadLine());
                    var newPlayer = new Player { PlayerId = id, PlayerName = name, PlayerAge = age };
                    team1.Add(newPlayer);
                    Console.WriteLine("Player is added successfully");
                    break;
                case 2:
                    Console.Write("Enter Player Id to Remove:");
                    int idToRemove = int.Parse(Console.ReadLine());
                    team1.Remove(idToRemove);
                    Console.WriteLine("Player is removed successfully");
                    break;
                case 3:
                    Console.Write("Enter Player Id:");
                    int idToGet = int.Parse(Console.ReadLine());
                    var playerById = team1.GetPlayerById(idToGet);
                    Console.WriteLine($"{playerById.PlayerId}\t{playerById.PlayerName}\t{playerById.PlayerAge}");
                    break;
                case 4:
                    Console.Write("Enter Player Name:");
                    string nameToGet = Console.ReadLine();
                    var playerByName = team1.GetPlayerByName(nameToGet);
                    Console.WriteLine($"{playerByName.PlayerId}\t{playerByName.PlayerName}\t{playerByName.Playe
rAge}");
                    break;
                case 5:
                    var allPlayers = team1.GetAllPlayers();
                    foreach (var player in allPlayers)
                    {
                        Console.WriteLine($"{player.PlayerId}\t{player.PlayerName}\t{player.PlayerAge}");
                    }
                    break;
            }
            Console.Write("Do you want to continue (yes/no)?:");
            string response = Console.ReadLine();
            if (response.ToLowerInvariant() == "yes")
            {
                goto start;
            }
            Console.ReadLine();
        }
```

```
    }
}
```