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

namespace PlayerandTeamPrjct
{
    interface ITeam
    {
        void AddPlayer(Player player);
        void RemovePlayer(int playerId);
        Player GetPlayerById(int playerId);
        List<Player> GetPlayersByName(string playerName);
        List<Player> GetAllPlayers();
    }

    // Player class
    class Player
    {
        public int PlayerId { get; set; }
        public string Name { get; set; }
        public int Age { get; set; }
        public int Id { get; internal set; }
    }

    // OneDayTeam class implementing ITeam interface
    class OneDayTeam : ITeam
    {
        private List<Player> players = new List<Player>();

        public void AddPlayer(Player player)
        {
            if (players.Count < 11)
            {
                players.Add(player);
                Console.WriteLine($"Player {player.Name} added to the team.");
            }
            else
            {
                Console.WriteLine("Cannot add more than 11 players to the team.");
            }
        }

        public void RemovePlayer(int playerId)
        {
            Player playerToRemove = players.FirstOrDefault(p => p.PlayerId ==
playerId);
            if (playerToRemove != null)
            {
                players.Remove(playerToRemove);
                Console.WriteLine($"Player {playerToRemove.Name} removed from the
team.");
            }
            else
            {
                Console.WriteLine("Player not found in the team.");
            }
        }

        public Player GetPlayerById(int playerId)
        {
            return players.FirstOrDefault(p => p.PlayerId == playerId);
        }

        public List<Player> GetPlayersByName(string playerName)
        {
            return players.Where(p => p.Name.Equals(playerName,
StringComparison.OrdinalIgnoreCase)).ToList();
        }

        public List<Player> GetAllPlayers()
        {
            return players;
        }
    }

    // Program class with the Main method
    class Program
    {
        static List<Player> players = new List<Player>();

        static void Main()
        {
            string continueOption = "yes";

            while (continueOption.ToLower() == "yes")
            {
                Console.WriteLine("Enter 1: To Add Player 2: To Remove Player by Id
3. Get Player By Id 4. Get Player by Name 5. Get All Players:");

                int choice;
                if (int.TryParse(Console.ReadLine(), out choice))
                {
                    switch (choice)
                    {
                        case 1:
                            AddPlayer();
                            break;
                        case 2:
                            RemovePlayerById();
                            break;
                        case 3:
                            GetPlayerById();
                            break;
                        case 4:
                            GetPlayerByName();
                            break;
                        case 5:
                            GetAllPlayers();
                            break;
                        default:
                            Console.WriteLine("Invalid choice. Please try again.");
                            break;
                    }
                }
                Console.Write("Do you want to continue (yes/no)?: ");
                continueOption = Console.ReadLine();
            }
        }

        static void AddPlayer()
        {
            Player player = new Player();
            Console.Write("Player Name: ");
            player.Name = Console.ReadLine();
            Console.Write("Player Id: ");
            player.Id = int.Parse(Console.ReadLine());
            Console.Write("Player Age: ");
            player.Age = int.Parse(Console.ReadLine());

            players.Add(player);

            Console.WriteLine($"{player.Name} is added successfully");
        }

        static void RemovePlayerById()
        {
            Console.Write("Enter Player Id to Remove: ");
            int playerId = int.Parse(Console.ReadLine());

            Player playerToRemove = players.Find(p => p.Id == playerId);
            if (playerToRemove != null)
            {
                players.Remove(playerToRemove);
                Console.WriteLine("Player is removed successfully");
            }
            else
            {
                Console.WriteLine("Player not found");
            }
        }

        static void GetPlayerById()
        {
            Console.Write("Enter Player Id: ");
            int playerId = int.Parse(Console.ReadLine());

            Player player = players.Find(p => p.Id == playerId);
            if (player != null)
            {
                Console.WriteLine($"{player.Id} {player.Name} {player.Age}");
            }
            else
            {
                Console.WriteLine("Player not found");
            }
        }

        static void GetPlayerByName()
        {
            Console.Write("Enter Player Name: ");
            string playerName = Console.ReadLine();
```

```
            Player player = players.Find(p => p.Name == playerName);
            if (player != null)
            {
                Console.WriteLine($"{player.Id} {player.Name} {player.Age}");
            }
            else
            {
                Console.WriteLine("Player not found");
            }
        }

        static void GetAllPlayers()
        {
            foreach (Player player in players)
            {
                Console.WriteLine($"{player.Id} {player.Name} {player.Age}");
            }
        }
    }
}
```