Source code:

Player and Team Requirements.

```
using System;
using System.Collections.Generic;

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

public interface ITeam
{
    void Add(Player player);
    void Remove(int playerId);
    Player GetPlayerById(int playerId);
    Player GetPlayerByName(string playerName);
    List<Player> GetAllPlayers();
}

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

    public OneDayTeam()
    {
        // Constructor to set capacity to 11
        oneDayTeam.Capacity = 11;
    }

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

    public void Remove(int playerId)
    {
        Player playerToRemove = oneDayTeam.Find(p => p.PlayerId == playerId);
        if (playerToRemove != null)
        {
            oneDayTeam.Remove(playerToRemove);
            Console.WriteLine($"Player {playerId} removed successfully.");
        }
        else
        {
            Console.WriteLine($"Player with ID {playerId} not found in the team.");
        }
    }

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

    public Player GetPlayerByName(string playerName)
    {
        return oneDayTeam.Find(p => p.Name.Equals(playerName,
StringComparison.OrdinalIgnoreCase));
    }

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

public class Program
{
    static void Main()
    {
        OneDayTeam oneDayTeam = new OneDayTeam();

        while (true)
        {
            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 = Convert.ToInt32(Console.ReadLine());

            switch (choice)
            {
                case 1:
                    Console.WriteLine("Enter Player Id:");
                    int playerId = Convert.ToInt32(Console.ReadLine());

                    Console.WriteLine("Enter Player Name:");
                    string playerName = Console.ReadLine();

                    Console.WriteLine("Enter Player Age:");
                    int playerAge = Convert.ToInt32(Console.ReadLine());

                    Player newPlayer = new Player { PlayerId = playerId, Name = playerName, Age =
playerAge };
                    oneDayTeam.Add(newPlayer);
                    break;

                case 2:
                    Console.WriteLine("Enter Player Id to Remove:");
                    int removePlayerId = Convert.ToInt32(Console.ReadLine());
                    oneDayTeam.Remove(removePlayerId);
                    break;

                case 3:
                    Console.WriteLine("Enter Player Id:");
                    int getPlayerById = Convert.ToInt32(Console.ReadLine());
                    Player playerById = oneDayTeam.GetPlayerById(getPlayerById);
                    if (playerById != null)
                        Console.WriteLine($"{playerById.PlayerId} {playerById.Name}
{playerById.Age}");
                    else
                        Console.WriteLine("Player not found.");
                    break;

                case 4:
                    Console.WriteLine("Enter Player Name:");
                    string getPlayerByName = Console.ReadLine();
                    Player playerByName = oneDayTeam.GetPlayerByName(getPlayerByName);
                    if (playerByName != null)
                        Console.WriteLine($"{playerByName.PlayerId} {playerByName.Name}
{playerByName.Age}");
                    else
```

```
                        Console.WriteLine("Player not found.");
                    break;

                case 5:
                    List<Player> allPlayers = oneDayTeam.GetAllPlayers();
                    foreach (var player in allPlayers)
                    {
                        Console.WriteLine($"{player.PlayerId} {player.Name} {player.Age}");
                    }
                    break;

                default:
                    Console.WriteLine("Invalid choice. Please try again.");
                    break;
            }

            Console.WriteLine("Do you want to continue (yes/no)?");
            string continueChoice = Console.ReadLine().ToLower();
            if (continueChoice != "yes")
                break;
        }
    }
}
```