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

namespace Project_1
{
    internal class Program
    {
        static void Main(string[] args)
        {
            FastPace_CricketAcademy team = new FastPace_CricketAcademy();

            while (true)
            {
                Console.WriteLine("Choose an option:");
                Console.WriteLine("1. Add Player");
                Console.WriteLine("2. Remove Player");
                Console.WriteLine("3. Get Player by Id");
                Console.WriteLine("4. Get Players by Name");
                Console.WriteLine("5. Get All Players");
                Console.WriteLine("6. Exit");

                if (int.TryParse(Console.ReadLine(), out int choice))
                {
                    switch (choice)
                    {
                        case 1:
                            Console.WriteLine("Enter Player Id:");
                            int playerId = int.Parse(Console.ReadLine());

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

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

                            team.AddPlayer(new Player { id = playerId, name = playerName, age = playerAge });
                            break;

                        case 2:
                            Console.WriteLine("Enter Player Id to remove:");
                            int playerIdToRemove = int.Parse(Console.ReadLine());

                            team.RemovePlayer(playerIdToRemove);
                            break;

                        case 3:
                            Console.WriteLine("Enter Player Id to get details:");
                            int playerIdToGet = int.Parse(Console.ReadLine());

                            Player playerById = team.getPlayerById(playerIdToGet);
                            if (playerById != null)
                            {
```

```
                    Console.WriteLine($"Player Found: Id: {playerById.id}, Name: {playerById.name}, Ag
e: {playerById.age}");
                }
                else
                {
                    Console.WriteLine("Player not found.");
                }
                break;

            case 4:
                Console.WriteLine("Enter Player Name to get details:");
                string playerNameToGet = Console.ReadLine();

                List<Player> playersByName = team.getPlayersByName(playerNameToGet);
                if (playersByName.Count > 0)
                {
                    Console.WriteLine($"Players Found with name '{playerNameToGet}':");
                    foreach (Player player in playersByName)
                    {
                        Console.WriteLine($"Id: {player.id}, Name: {player.name}, Age: {player.age}");
                    }
                }
                else
                {
                    Console.WriteLine($"No players found with name '{playerNameToGet}'.");
                }
                break;

            case 5:
                List<Player> allPlayers = team.getAllPlayers();
                Console.WriteLine("All Players:");
                foreach (Player player in allPlayers)
                {
                    Console.WriteLine($"Id: {player.id}, Name: {player.name}, Age: {player.age}");
                }
                break;

            case 6:
                Console.WriteLine("Exiting the program.");
                Environment.Exit(0);
                break;

            default:
                Console.WriteLine("Invalid choice. Please choose correct option.");
                break;
        }
    }
    else
    {
        Console.WriteLine("Invalid input. Please enter a valid number.");
    }

    Console.WriteLine();
  }
}
```

```
    }
}
```

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

namespace Project_1
{

        public class Player
        {
            public int id { get; set; }
            public string name { get; set; }
            public int age { get; set; }
        }

}
```

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

namespace Project_1
{
    internal interface Team
    {
        void addPlayer(Player player);
        void removePlayer(int playerId);
        Player getPlayerById(int playerId);
        List<Player> getPlayerByName();
        List<Player> getAllPlayers();
    }
}
```

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

namespace Project_1
{
```

```
   public class FastPace_CricketAcademy
   {
      private List<Player> playerList;
      public FastPace_CricketAcademy()
      {
         playerList = new List<Player>();
      }
      public void AddPlayer(Player player)
      {
         if (playerList.Count < 11)
         {
            playerList.Add(player);
            Console.WriteLine("Player {0} added to the team", player.name);
         }
         else
         {
            Console.WriteLine("Cannot add player because the team already have 11 players");
         }
      }
      public void RemovePlayer(int playerId)
      {
         Player playerToBeRemoved = playerList.FirstOrDefault(p => p.id == playerId);
         if (playerToBeRemoved != null)
         {
            playerList.Remove(playerToBeRemoved);
            Console.WriteLine("Player {0} is removed from team", playerToBeRemoved.name);
         }
         else
         {
            Console.WriteLine("Cannot find the player with Id {0}", playerId);
         }
      }
      public Player getPlayerById(int playerId)
      {
         return playerList.FirstOrDefault(p => p.id == playerId);
      }
      public List<Player> getPlayersByName(string playerName)
      {
         return playerList.Where(p => p.name.Equals(playerName, StringComparison.OrdinalIgnoreCase))
.ToList();
      }
      public List<Player> getAllPlayers()
      {
         return playerList.ToList();
      }
   }
}
```