**Title: Player and Team Requirements.**

# **Introduction:**

The Player and Team Management Console Application is designed to facilitate the management of a one-day cricket team. The application comprises a console app and a class library, each serving a distinct purpose. The console app allows users to interactively add, remove, and retrieve player information from the team, while the class library provides the underlying structure for player and team management.

# **Namespace and File Structure:**

The solution is organized under the "PlayerAndTeamApp" namespace for the console app and "PlayerAndTeamLibrary" namespace for the class library. The structure includes the following files:

## **Program.cs (PlayerAndTeamApp):**

* The main entry point for the console application.
* Utilizes the functionalities provided by the "OneDayTeam" class in the class library.
* Implements a user-friendly menu system for adding, removing, and retrieving player information.

## **Player.cs (PlayerAndTeamLibrary):**

* Represents the Player entity with properties for PlayerId, PlayerName, and PlayerAge.
* Serves as the basic building block for player information.

## **ITeam.cs (PlayerAndTeamLibrary):**

* Defines the "ITeam" interface, outlining the methods for adding, removing, and retrieving player information.
* Enforces a contract that any team class should implement these methods.

## **OneDayTeam.cs (PlayerAndTeamLibrary):**

* Implements the "ITeam" interface, providing functionality for managing a one-day cricket team.
* Includes methods for adding, removing, and retrieving player information based on PlayerId and PlayerName.
* Enforces a team size limit of 11 players.

# **Functionality:**

The console app's main functionality revolves around a user-driven menu system, where the user can choose from options such as adding a player, removing a player by ID, retrieving player details by ID or name, and displaying the entire team roster. The app ensures a seamless and intuitive experience by prompting the user for input and providing appropriate feedback.

# **User Interaction:**

## **Adding a Player:**

Users input PlayerId, PlayerName, and PlayerAge, creating a new player entity that is added to the team roster.

## **Removing a Player:**

Users enter the PlayerId of the player they wish to remove, prompting the application to search and eliminate the player from the team.

## **Retrieving Player Details:**

Users can retrieve player details either by ID or name, with the app providing the corresponding information or notifying if the player does not exist.

## **Displaying All Players:**

The app presents a comprehensive list of all players in the team, showcasing their PlayerId, PlayerName, and PlayerAge.