Driver & Interface Adapter

SportsApp

Parent: None
Subclasses: None
Layer: Framework & Drivers

- Starting point of the app
- Prints starting instructions for the user to follow
- Accepts input from the user and passes it to the CommandManager
- Prints out the output received from the CommandManager

Relationships:

- CommandManager

CommandManager

- Parses the input string into a command and its arguments
- Identifies the command and passes it to an appropriate object

Parent: None
Subclasses: None
Layer: Interface Adapter

Relationships:

- InputParser
- PlayerStatManager
- PlayerStatComparer
- PlayerStatPredictor
- TeamStatManager
- TeamStatComparer
- TeamStatPredictor
- LeagueMemberManager

InputParser

Parent: None
Subclasses: None
Layer: Interface Adapter

- The class is responsible for parsing the user's input and extracting the arguments out of the input
- It will store the keyword command and the arguments for the command separately

- CommandManager

Related to Teams

TeamStatManager

Parent: None
Subclasses: None
Layer: Use Case

- Find or compute statistics about a given Team

- TeamManager
- Team

TeamStatComparer

Parent: None
Subclasses: None
Layer: Use Case

- Compare two or more Teams on a given statistic

- TeamStatManager

TeamStatPredictor

Parent: None
Subclasses: None
Layer: Use Case

- Given a Team has a past record for a statistic, predict their future performance on that statistic

- TeamStatManager

Team (Abstract)

Parent: None
Subclasses: HockeyTeam, TennisTeam
Layer: Entity

- Store a team's name, home city, players, number of games played, number of wins, number of losses, and number of ties
- Getters and Setters for above

- None

HockeyTeam

Parent: Team
Subclasses: None
Layer: Entity

- Store the information specified in team class

- TeamManager

TennisTeam

Parent: Team
Subclasses: None
Layer: Entity

- Store the information specified in team class

- TeamManager

TeamManager

Parent: None
Subclasses: None
Layer: Entity

- Store Teams, with getter (for Use Cases to resolve argument name into Team object)
- Create and record new Teams

- Team

Related to Members & Betting

LeagueMemberManager

Parent: None
Subclasses: None
Layer: Use Case

- Create and record the Members in the fantasy league
- Create and record the ongoing Matches
- Notify stored Matches when a Member bets on them or when their outcome is resolved

- Member
- Match

LeagueMember

Parent: None
Subclasses: None
Layer: Entity

- Represent a Member of a fantasy league, who bets on games
- Stores the Member’s name
- Tracks the amount of matches they have predicted correctly and incorrectly

- None

Match

Parent: None
Subclasses: None
Layer: Entity

- Store the two teams who are competing in the match
- Getters and setters for above
- Record and store which Members have bet on which outcomes of the match
- After the match ends, update players who bet correctly and who bet incorrectly

- MemberManager

Related to Players

PlayerStatManager

Parent: None
Subclasses: None
Layer: Use Case

- Report a given stat (or all stats) for a hockey player in one or more seasons
- Stats that can be reported:
  - See HockeyPlayer card (any of the information being stored by HockeyPlayer can be reported)

- HockeyPlayer
- PlayerList

TennisPlayerStatManager

Parent: None
Subclasses: None
Layer: Use Case

- Return the value of a stat (or all stats), given a tennis player's name, a tournament name, and a stat
- Stats that can be reported are:
  - See TennisPlayer card (any of the information being stored by TennisPlayer can be reported)

- TennisPlayer
- TennisPlayerList

PlayerStatComparer

Parent: None
Subclasses: None
Layer: Use Case

- Compare two or more hockey players on a given statistic in a specific season
- Stats that can be compared:
  - number of games played
  - number of goals
  - number of assists
  - number of points
  - shooting percentage
  - number of shots

- HockeyPlayer
- PlayerList

TennisPlayerStatComparer

Parent: None
Subclasses: None
Layer: Use Case

- Compare two tennis players who participated in a competition based on a given stat
- Stats that can be compared are:
  - number of aces
  - number of double faults
  - number of serve points
  - number of first serves
  - number of break points saved

- TennisPlayer
- TennisPlayerList

PlayerStatPredictor

Parent: None
Subclasses: None
Layer: Use Case

- Given a hockey player's name and a stat, predict their future stat using linear regression
- Stats that can be predicted are:
  - number of games played
  - number of goals
  - number of assists
  - number of points
  - shooting percentage
  - number of shots

- HockeyPlayer
- PlayerList

TennisPlayerStatPredictor

Parent: None
Subclasses: None
Layer: Use Case

- Given a tennis' player's name and stat, predict their future stat with linear regression
- Stats that can be compared are:
  - number of aces
  - number of double faults
  - number of serve points
  - number of first serves
  - number of break points saved

- TennisPlayer
- TennisPlayerList

Player (Abstact)

Parent: None
Subclasses: HockeyPlayer
Layer: Entity

- Store player’s name
- Getter and setter for above

- None

HockeyPlayer

Parent: HockeyPlayer
Subclasses: None
Layer: Entity

- Store the season, position, number of games played, number of goals, number of assists, number of points, shooting percentage, number of shots, and skater shoots

- None

TennisPlayer

Parent: Player
Subclasses: None
Layer: Entity

- Store a tennis player's:
  - age
  - nationality (represented by the 3 letter IOC code for their country)
  - number of aces
  - number of double faults
  - number of first serves
  - number of serve points
  - number of break points saved
- Getter and setters for above

- None

PlayerList

Parent: None
Subclasses: None
Layer: Entity

- Read a .csv file containing data about hockey players for multiple seasons
- Create a map with a season as the key, and a list of hockey players who played during that season

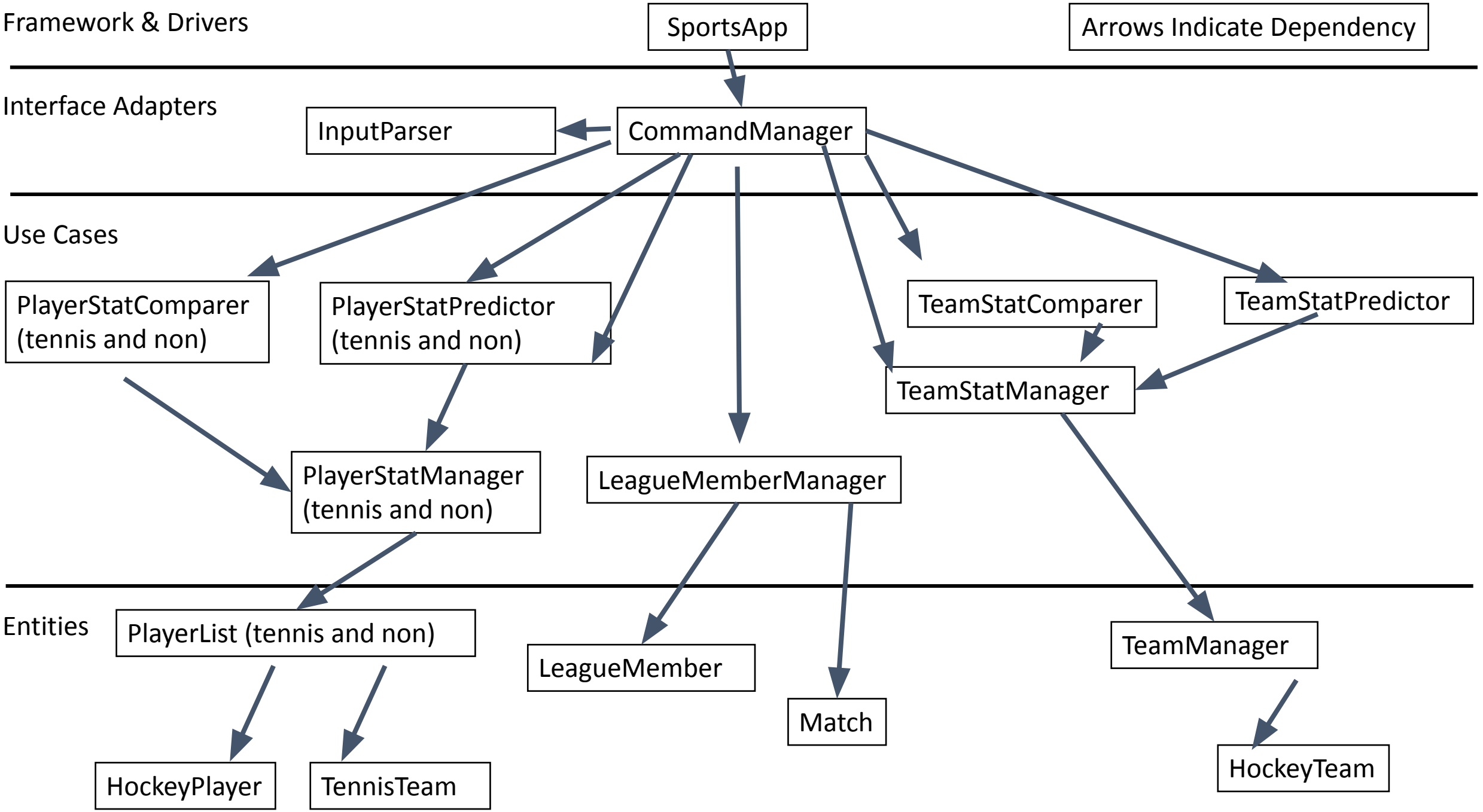
- HockeyPlayer

TennisPlayerList

Parent: None
Subclasses: None
Layer: Entity

- Read a .csv file containing data about tennis players that participated in different tournaments
- Create a map where the keys are competition names and the values are a list of TennisPlayer objects
- Search for a tennis player in a competition

- TennisPlayer


Framework & Drivers
SportsApp
Arrows Indicate Dependency
Interface Adapters
InputParser
CommandManager
Use Cases
PlayerStatComparer (tennis and non)
PlayerStatPredictor (tennis and non)
TeamStatComparer
TeamStatPredictor
TeamStatManager
PlayerStatManager (tennis and non)
LeagueMemberManager
Entities
PlayerList (tennis and non)
LeagueMember
Match
TeamManager
HockeyPlayer
TennisTeam
HockeyTeam