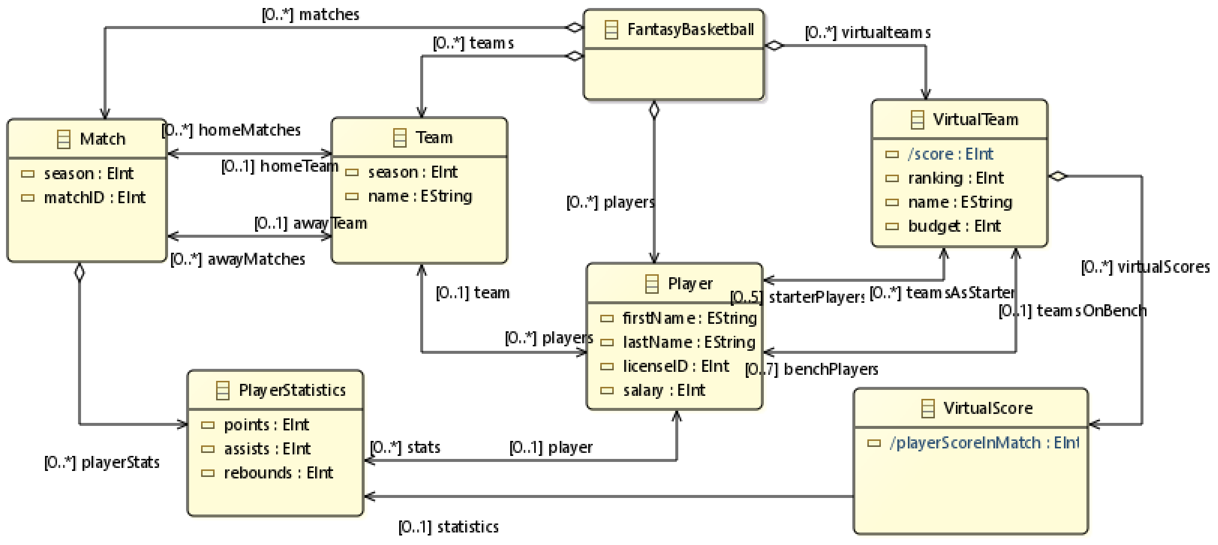
Domain Modeling with Umple



(The black diamonds of containment associations are only lost at export ☺)
Note the following key abstractions in this domain model:

- Team and VirtualTeam should be two separate classes. A Team represents a specific Team within a season, i.e. multiple (real) teams will exist under the same name for different seasons.
- Either both classes are connected to the same set of Players. Alternatively, class Player could be split into two: RealPlayer and VirtualPlayer when (1) containment of Players can be assigned to Teams and VirtualTeams respectively, and (2) make the role of the player to become an enumeration attribute (with values *starterPlayer* / *benchPlayer*).
- The sample solution merge classes User and VirtualTeam under the assumption that each user may have a single Virtual Team. If you dropped this assumption, then you may have a separate User class.
- The season attribute should be noted for Matches and Teams (and optionally also for VirtualTeams) since a player may play in different teams in different seasons. One may introduce a separate Season class, and then the corresponding attribute should become an association.
- PlayerStatistics (i.e. the real statistics during a game) and their VirtualScore (for a VirtualTeam) need to be separated (in one way or the other).
- You may abstract from *homeTeam* and *awayTeam* and have a single association called *teams.*
- You may introduce an extra reference from VirtualScore to Player to improve navigability (which may be a derived reference in principle).
- You may decide to use * multiplicities for *starterPlayers* and *benchPlayers* (although it is slightly less precise).
- All (instance level) elements are aligned into containment hierarchy (see containment references between classes). There are multiple options for achieving such containment hierarchy.
- You may define extra navigability for association ends compared to the sample solution.
- Note the multiplicities. All containment references should have 0..* at one end and 1 at the container end.

- No domain classes of list, container, etc.

The corresponding Umple code is:

```
namespace ca.mcgill.ecse321.fantasy;

class FantasyBasketball {
       1 <@>- * Team teams;
       1 <@>- * VirtualTeam virtualTeams;
       1 <@>- * Match matches;
       1 <@>- * Player players;
}

class Player {
       firstName;
       lastName;
       Integer licenseId;
}

class Team {
       Integer season;
       name;
       * teams -- * Player players;
}

class VirtualTeam {
       * teamsAsStarter -- 0..5 Player starterPlayers;
       * teamsOnBench -- 0..7 Player benchPlayers;
       1 <@>- * VirtualScore virtualScores;
       Integer score;
       Integer ranking;
       name;
}

class Match {
       Integer season;
       autounique matchId;
       * -> 1 Team homeTeam;
       * -> 1 Team awayTeam;
       1 <@>- * PlayerStatistics playerStats;
}

class PlayerStatistics {
       * -- 1 Player player;
       Integer points;
       Integer assists;
       Integer rebounds;
}

class VirtualScore {
       * -> 1 PlayerStatistics stat;
       Integer playerScoreInMatch;
}
```