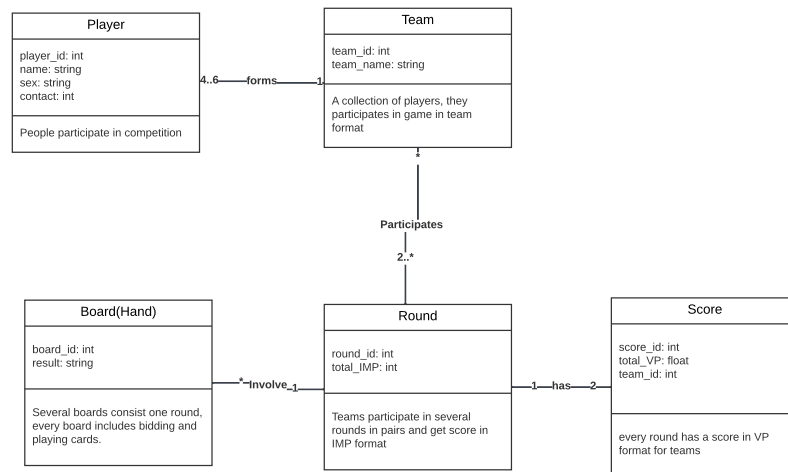**Project :** Bridge Competition Management System

1. The system aims to develop a monitor that could help anticipants and spectators of a Bridge competition find relative information quickly especially regarding live scores, and detailed performance statistics.

Nouns & Actions

Rules:

1. Players must register for the competition, providing their personal details and contact information.
2. Players form teams often with regular partners (in pairs). A team contains 4 or 6 players so that there are 2 or 3 pairs.
3. The competition consists of multiple rounds which teams all participate in, each involving a set number of boards (hands) to be played.
4. At the beginning of each round, cards are randomly distributed to players according to the rules of the game. In the big tournaments, the machine deals the cards.
5. After every round, system shows real-time scores for teams, enabling participants and spectators to follow the progress of the competition.
6. Teams engage in bidding to determine the contract and play the hands according to the contract's specifications.
7. International Match Point (IMP) scores are calculated based on the results of each round, taking into account the difference in performance between competing teams.
8. Victory Points (VP) are computed to establish rankings among teams in the competition.

2. UML:


```
┌─────────────────────────┐              ┌─────────────────────────┐
│         Player          │              │          Team           │
├─────────────────────────┤              ├─────────────────────────┤
│ player_id: int          │              │ team_id: int            │
│ name: string            │ 4..6  forms 1│ team_name: string       │
│ sex: string             │──────────────│                         │
│ contact: int            │              ├─────────────────────────┤
├─────────────────────────┤              │ A collection of players,│
│ People participate in   │              │ they participates in    │
│ competition             │              │ game in team format     │
└─────────────────────────┘              └─────────────────────────┘
                                                     │ *
                                                Participates
                                                     │ 2..*
┌─────────────────────────┐   ┌─────────────────────────┐   ┌─────────────────────────┐
│      Board(Hand)        │   │          Round          │   │          Score          │
├─────────────────────────┤   ├─────────────────────────┤   ├─────────────────────────┤
│ board_id: int           │ * │ round_id: int           │ 1 │ score_id: int           │
│ result: string          │ Involve 1│ total_IMP: int   │ has 2│ total_VP: float      │
│                         │──────────│                  │──────│ team_id: int         │
├─────────────────────────┤   ├─────────────────────────┤   ├─────────────────────────┤
│ Several boards consist  │   │ Teams participate in    │   │ every round has a score │
│ one round, every board  │   │ several rounds in pairs │   │ in VP format for teams  │
│ includes bidding and    │   │ and get score in IMP    │   │                         │
│ playing cards.          │   │ format                  │   │                         │
└─────────────────────────┘   └─────────────────────────┘   └─────────────────────────┘
```

3. Data Structure and Functionality:

Functionalities:

1. Rank of teams: A ranking given based on each team's score;
2. Succeed in contract: when the team make the contract (like get "6D+1" or "3NT=") it got one point, sum up all the points for each team and store it;

Data Structure:

1. Rank of teams: use a Redis sorted set with key, "RankOfTeams:teamId" and "score" in "team" as value.
2. Succeed in contract: use a Redis set with key, "SucceedInContract:teamId" and sum of numbers of "+" and "=" occurs in "result" under "rounds" of "team" as value. (Further thought, when a team have a high number of successful contracts but still get low rank, it means a team is too conservative to strive for a higher contract)