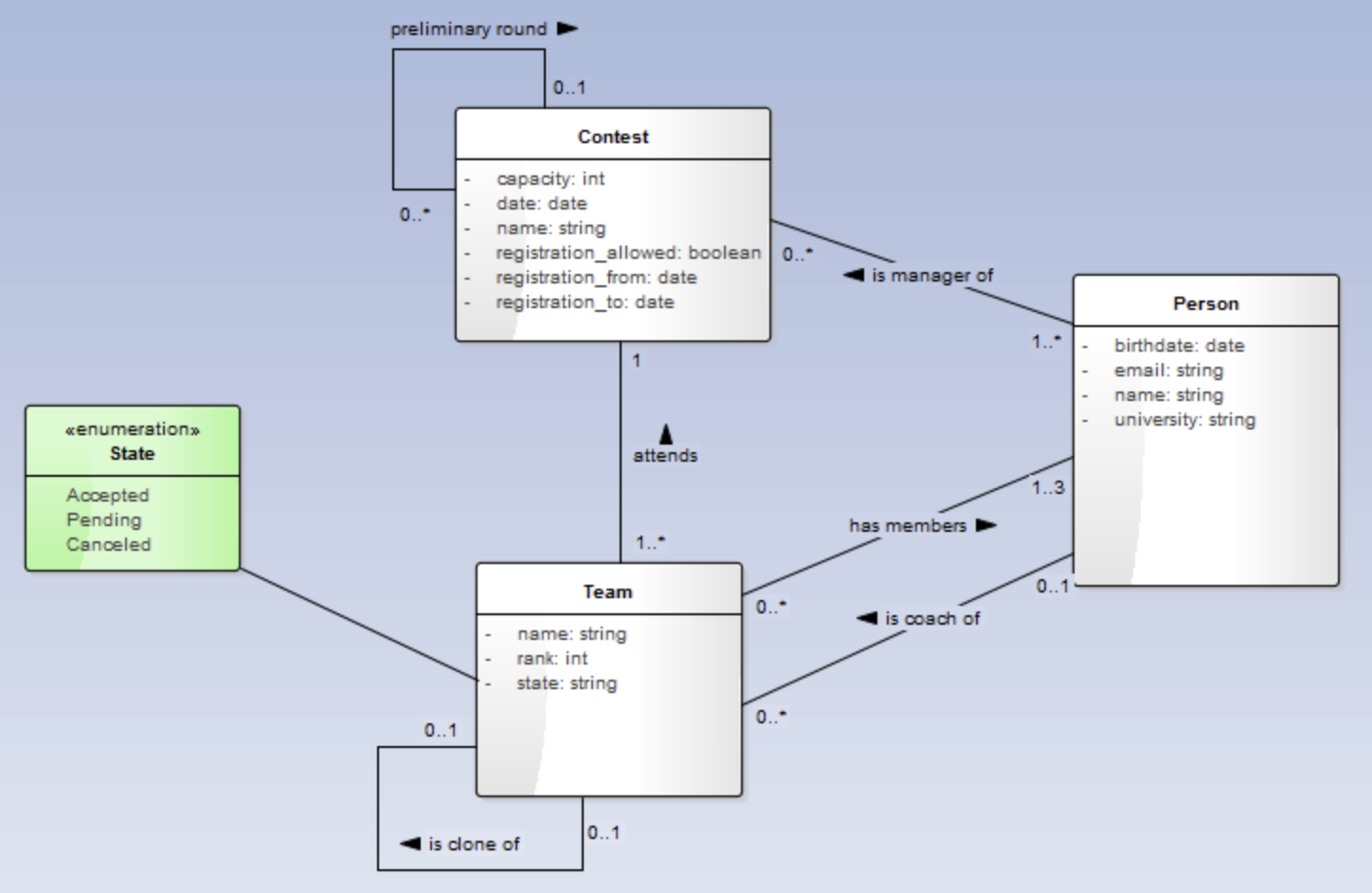
* It's a system for teams to register for contests.
* The system class diagram should reflect this:



Apply your JPA knowledge to implement the model below in Figure 1.

1. There should be contests, sub-contests, and 3 teams in sub-contest, each with 3 contestants and one coach, add contest manager).
2. There should be APIs to select all teams in the given contest and print them.
3. There should be API to get a report stating how many students there are grouped by age.
4. There should be an API to get the results after calculating the current contest occupancy and comparing it with the capacity of the contest.
5. These jobs should maintain 3 business rules specific described below:
   1. contest registration(Long contestId, Team team) is allowed only for teams.
      1. with 1 coach and 3 team members that are distinct from each other,
      2. only when the contest has available capacity,
      3. only when the team members are younger than 24 years,
      4. team members cannot be on a different team in the contest.
      5. TEAM has no ID
      6. TEAM members are specified by ID or are new!
      7. TEAM status defaults to PENDING not matter what, rank is ignored
   2. team/contest changes are allowed only if the contest indicates it is writable (**notEditable**) - flag (a boolean field).
   3. we can promote a team if there is **enough space** in the super contest and the **team rank is 1-5**. **We need again to check the rules from bullet 1**.

<https://github.com/tonnidas/HallOfFameGenerator>

https://github.com/tonnidas/ContestRegistrationService