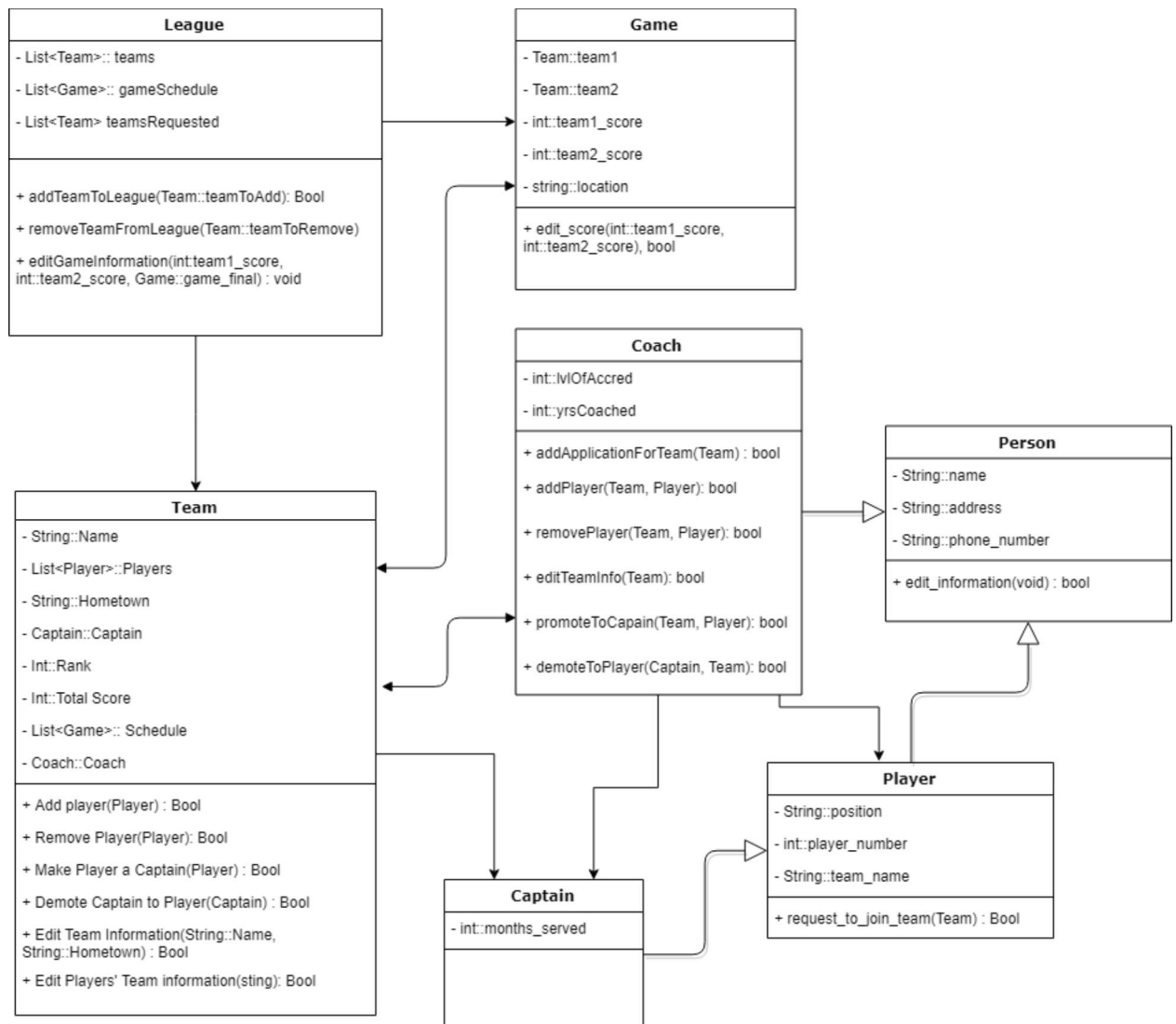


Reading Assignment #2

1. For this problem we should use a bottom up design approach (pg. 112). The application is very complex and abstract, especially the main feature, which is image comparison. This problem is so difficult on it's own that it is most efficient to work on small complex tasks, that solve the core of the application, and then build the rest of the subsystem complex around that. It would also benefit us greatly to make use of experimental prototyping. Since it is unclear how or how well the main image recognition and search feature would work, it would be great to design some prototypes before designing the rest of the program (pg.114). Doing this will give us performance standards and other requirements that the rest of the system will have to meet to work with the complex image search system. Since the image recognition system is so complex on it's own, and the entire program is designed around it, it is essential that we maintain low coupling in the system (pg. 100). We need to be able to test and modify the complex subsystems that run our image recognition and search feature easily. If coupling is high between subsystems, then it would make it impossible to maintain such a complex system down the road.
2. Source control is critical to our term project, it is difficult to work around all of our group members schedules to actually have team meetings often. Using git reduces that complication and avoids running into the problems Joel mentions. Joel poses two questions regarding bugs, "Do you have a bug database?" and "Do you fix bugs before writing new code?". So far in our term project we have gone with the approach of fixing bugs before writing new code, I agree with it simply removes complications later in development and is time efficient to deal with bugs with the code fresh on your mind. Having a bug database seems extremely useful for our project, so far we have just been able to fix the bugs and push code with 0 bugs found. When we come across bugs that we do not have time to fix in the same day, we should follow Joel's advice on creating a table to store everything about it. Uploading it to a group google doc so it is public knowledge within the group and everybody can have all the information known about bugs and keep it fresh on everybody's mind will allow for easy tracking of bugs. "Do you have an up-to-date schedule?", this is an important factor to a group school project. All of us have different classes, due dates, classes, etc. So keeping everybody on a clear cut schedule so we are not scrambling to get our code submitted on the last day is important, this is also something we have slacked on so far. We have been following Sakire's schedule included in the project assignment, which is okay but we still get very pressured by the end of the due date to get everything done, we need an up-to-date weekly schedule to keep everybody on top of the project.

Reading Assignment #2

3.



Assumptions:

We are assuming the coach can manage multiple teams, adding, removing players, promoting and demoting captains as well as changing team names, hometowns. We assume much of the rankings, time served as captain, and other variables and features are maintained automatically without human interaction. This means that when a Game instance is updated, the teams that competed in the game will update their rank, and total score automatically.