Status Summary - Enock Yared

I have almost 60-80% of my project completed as of right now. A lot has changed in my initial approach as it was very hard to use Jsoup and other tools to web scrap live data. Instead, I have gone with something similar that we have worked on but still relating to NBA statistics. I am using NBA2k ratings and simply reading a prepared file into my game engine. From there the user has an interface in which they can either use data or generate a simulated league with players that are randomly created and attributes chosen randomly.

Builder Pattern: I've used this in my `Team` and `Game` classes, which helps me build objects step by step, making the code more readable and maintainable.
Singleton Pattern: I could apply this to my `League_NBA` and `Playoffs` classes to ensure there's only one instance, which would help in managing league or season data consistently.
Composite Pattern: I see potential in using this for managing `Player` objects within `Team`, allowing me to treat individual players and teams uniformly.
Strategy Pattern: I've implied this in my `GameSimulation`, which allows me to switch between different game simulation strategies within the `Game` class.
Observer Pattern: I could implement this for real-time updates on game states, making the `Game` class a subject that notifies observers like scoreboards or statistics trackers.
Factory Method: I'm considering using this to simplify the creation of `Game` or `PlayoffComp` objects, especially since their instantiation process can be complex.

//TODO//
-JUnit Testing
- Finish debugging and get the first interface completed
- Get Auto Generated Season Completed
- Go back and scale up to all 30 teams for Option 1
-Incorporate more OOAD

//STRETCH GOALS//
- Include updated data if possible
- Injury probability reports
- Home/Away Advantages