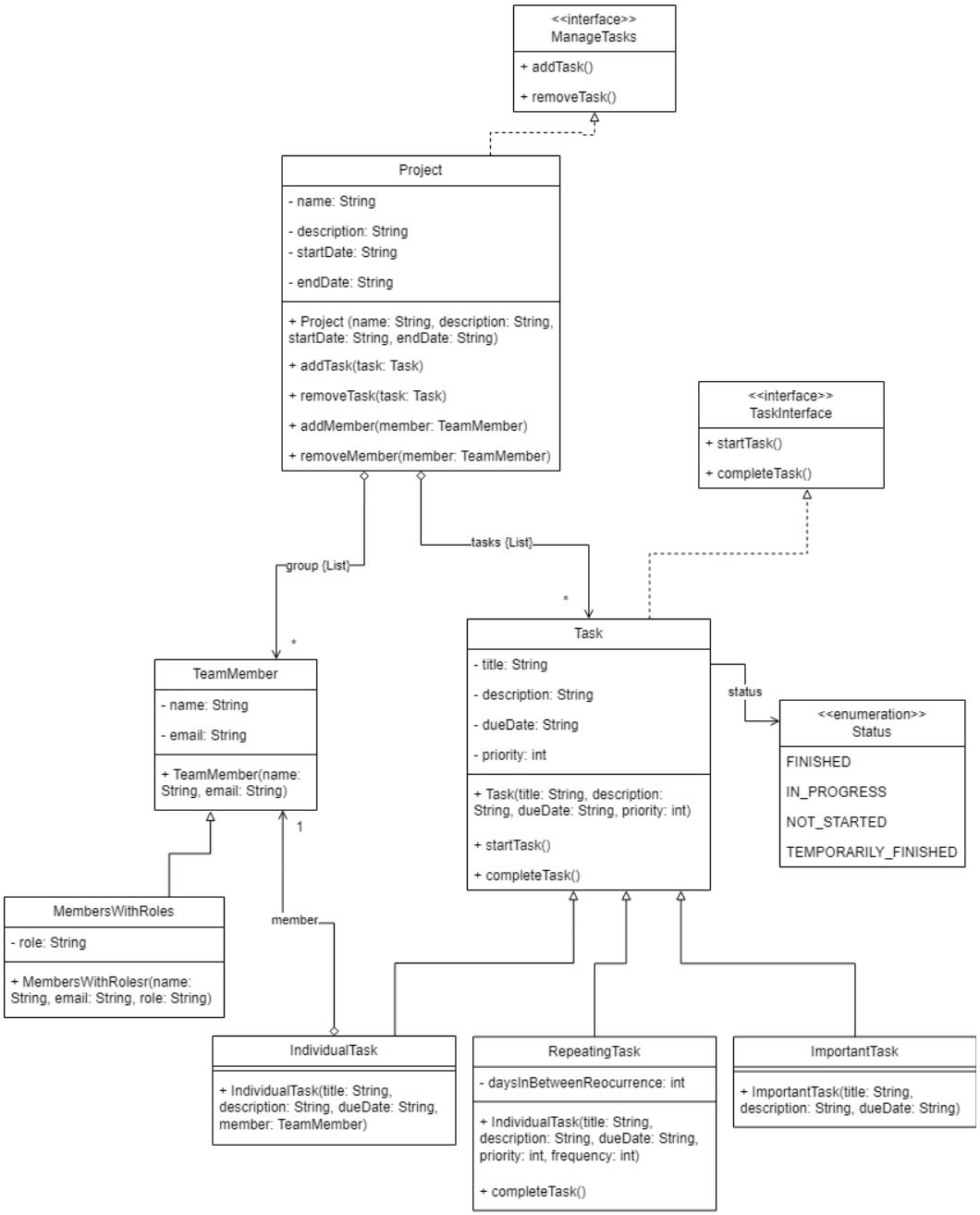
CS 2340 SOLID & GRASP Group Activity Writeup

Participating Members: Aayush Anantha, Blake Berman, Euan Ham, Beckham Lam, Nityam Mulani, & James De Ocampo

Class Diagram:



Principles:

1. Interface Segregation Principle: The code follows interface segregation principle as all the interfaces involved in the code are very specific with unique methods. This allows classes implementing these interfaces to later be able to implement other interfaces as well, giving flexibility that would not be present without separating of different interfaces.
2. Single Responsibility Principle: The code follows this principle as each class has its own responsibility. As an example, the task class specifically has the role of managing task-related attributes and methods. The RepeatingTask class has its own separate role, extending class, and adding its own functions specific to repeated tasks that normal ones may not have.
3. Open/Closed Principle: Again with the Task example, while Task and TaskInterface are immutable at this point, if there was the need to have a new type of task such as LongTermTask, then the original files would not need to be modified. Instead, we would follow the open closed principle and just add onto what already exists.
4. Liskov Substitution Principle: The repeating tasks or high importance tasks instances are completely qualified to substitute for regular task instances, which is the parent class of the aforementioned two classes. This follows the Liskov Substitution Principle as it states that child replacement of parent classes should not affect the correctness of the program.
5. Creator: The class that aggregates another class is responsible for making instances of the other class; for the instance in this project, the Project class contains the TeamMember and Task instances and therefore also has the responsibility of creating those instances.
6. Information Expert: The RepeatingTask class has the information required to figure out the recurrence frequency, so it makes sense that the method to compute the frequency is also in the same RepeatingTask class.