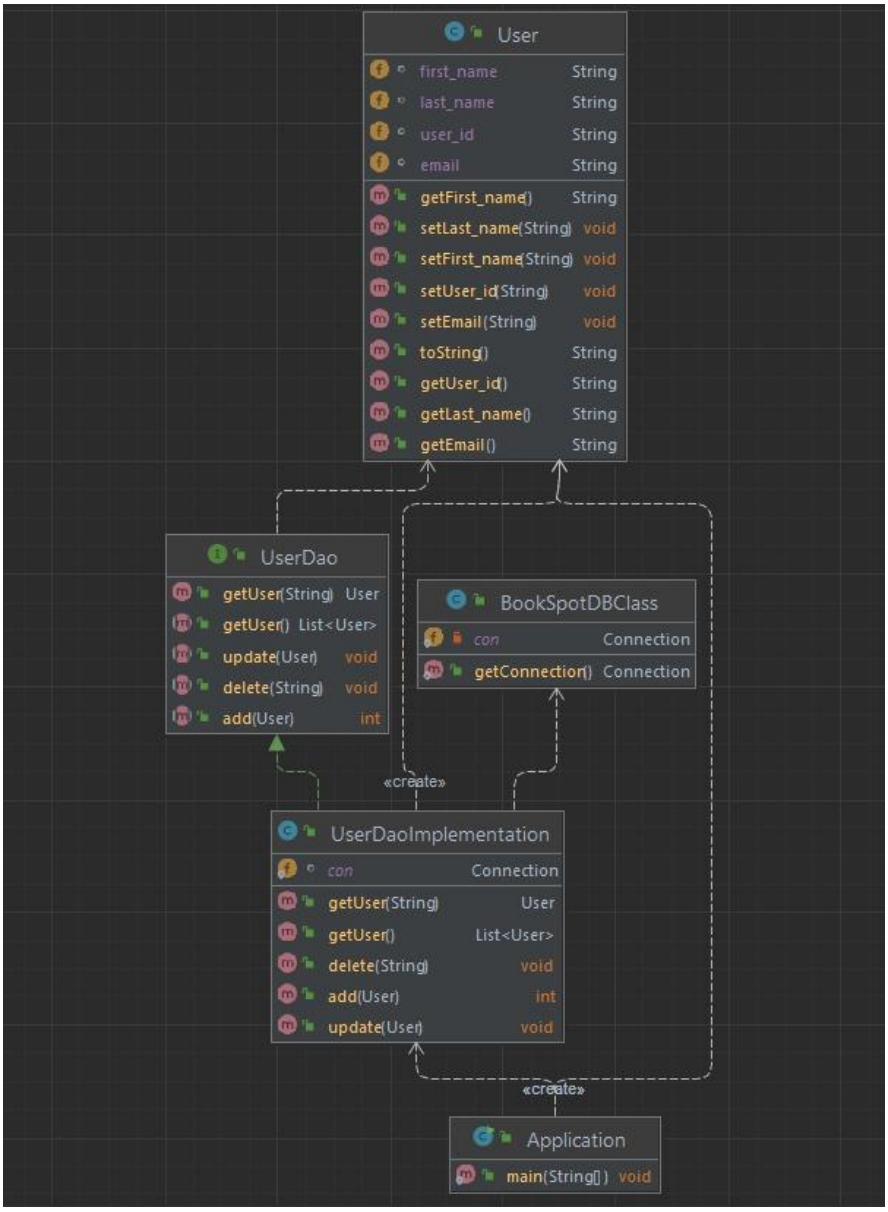
#575 OOAD: Assignment 2.1 – Group 1

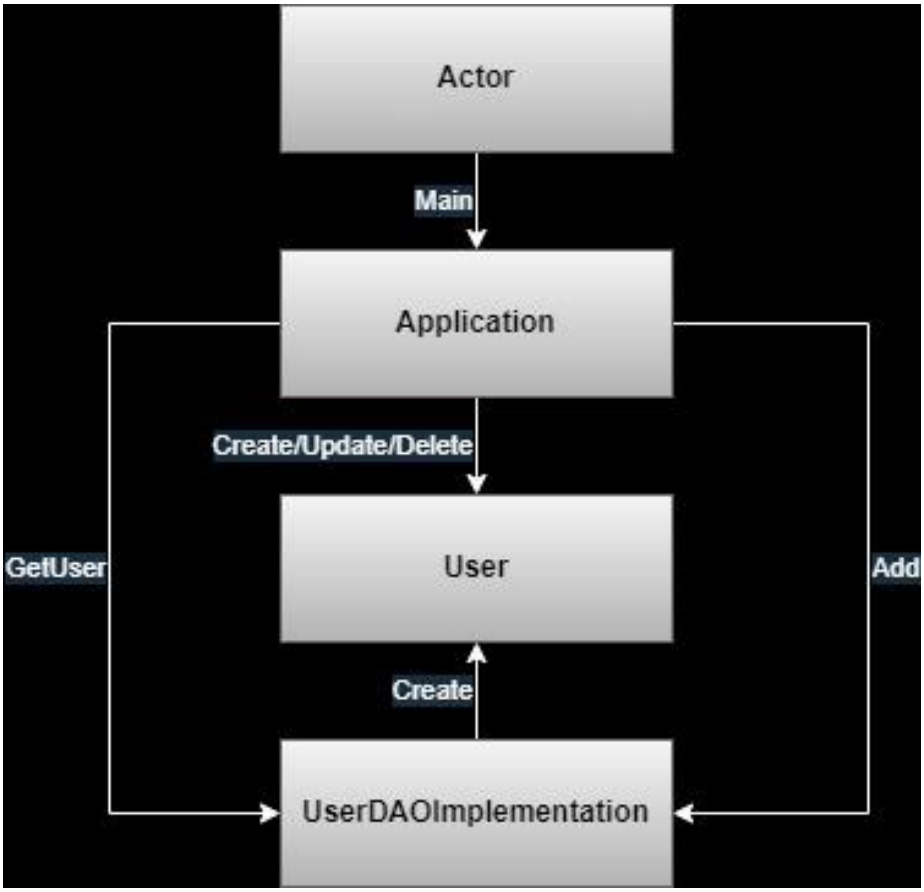
1. **Why singleton should be applied in the scenario and how it improves your design?**

- The Singleton's objective is to limit the number of objects created while giving for the flexibility to produce more if the scenario changes. As there is only one Singleton instance, any instance fields of a Singleton, including static fields, will only appear once per class.
- Singleton prohibits other things from creating duplicates of the Singleton object, guaranteeing that all objects have access to the same instance. As the class is in charge of the instantiation process, it has the ability to alter it at any time.
- When just one instance of a class is required to control the action throughout the execution, this method is always preferred.
- It accelerates the development process by using a tried-and-true development methodology. For developers and architects who are familiar with these patterns, it also increases readability.
- Singleton Design patterns are generic solutions that are written in a way that does not need information about the situation at hand.
- Singleton ensures that a class has only one instance and gives it with global access. The singleton patten is encapsulated on its initial usage.
- In each scenario and at any cost, a singleton class should not contain numerous instances. For logging, driver objects, caching and thread pools, and database connections, singleton classes are employed.
- Using a Singleton Pattern, we built a Bookstore Database with a table containing User entries. This also creates a java application that will connect with the Bookstore Database. A singleton class will create the connection and include all the essential driver manager information. Using a model object of User, we will conduct data manipulation actions such as add, delete and update from the Java program.
- The Singleton Pattern helped in saving memory usage in this Database connection as the objects were not created at every request.

A. Class Diagram for Singleton Pattern:



B. Collaboration Diagram for Singleton Pattern:


Actor
Main
Application
Create/Update/Delete
User
GetUser
Add
Create
UserDAOImplementation