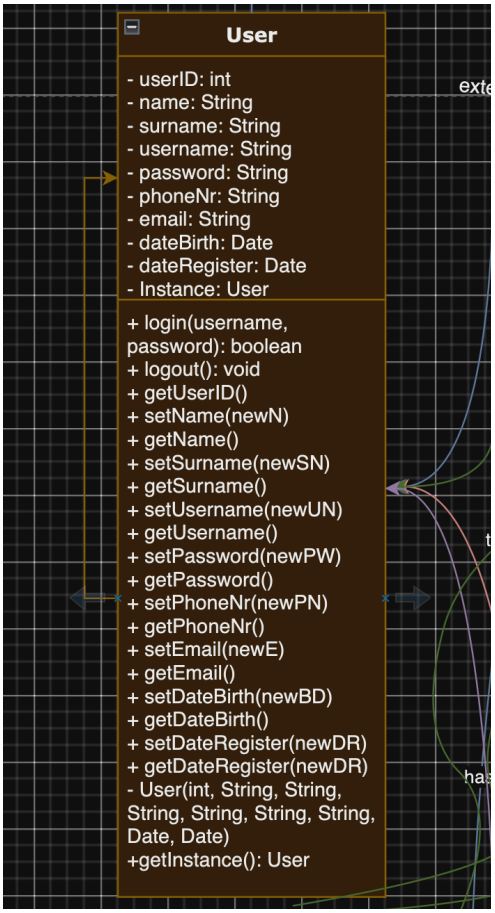
Design Pattern present in our project.

Singleton.

The singleton is a fundamental design pattern, which belongs to the creational design patterns. It is used to prevent the instantiation of more than one object from a particular class.

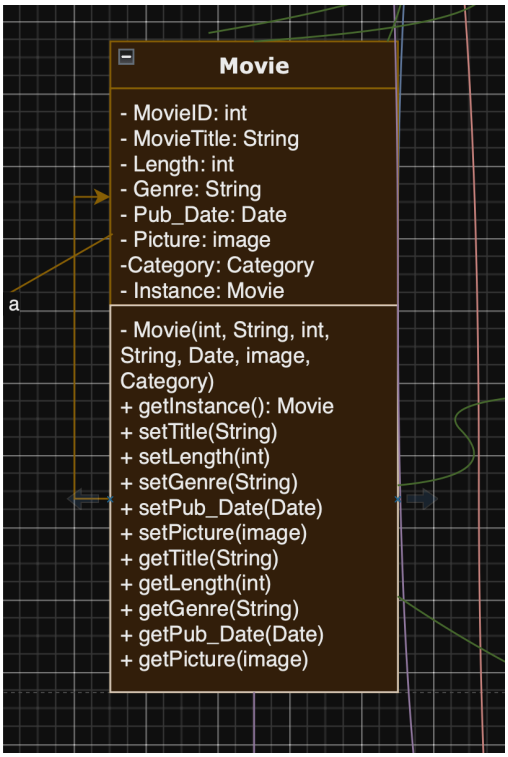
There are several benefits of using this particular design pattern, from which we can mention the added efficiency of sparing the memory from overloading with several instances of the same object. In our case, only one instance of the following classes needs to be created and used.

To achieve this, the following classes contain an object of themself, and the constructor which initiates the objects is declared as private. Instead of directly calling the constructor from outside of the class, which would lead to the creation of various objects from it, the constructor is only called once inside the getInstance method, only if it has not previously been done.



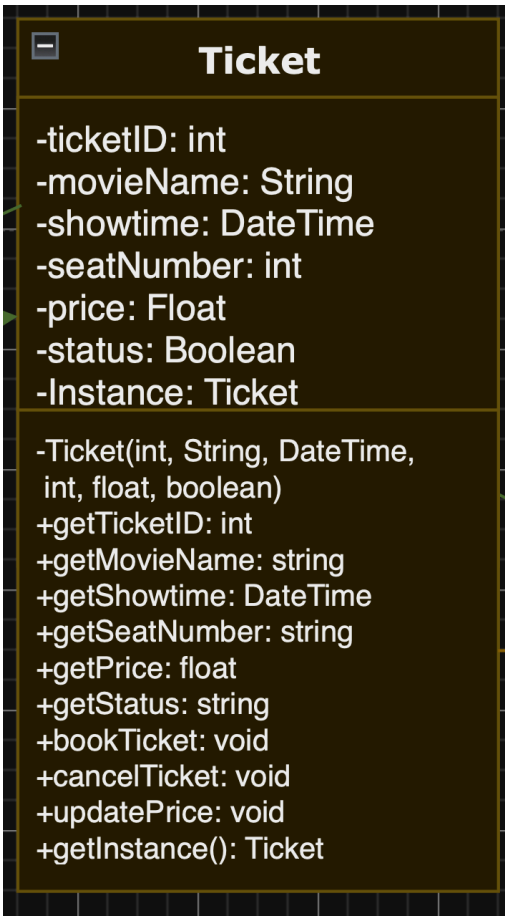
User Class (This extends to all of its children classes: Admin, Manager, Client, Staff)

As you see the class has a private constructor and a public getInstance function that checks if the instance of the class is created or not. If yes, return it, else create a new one.



Movie Class.

As you see the class has a private constructor and a public getInstance function that checks if the instance of the class is created or not. If yes, return it, else create a new one.



Ticket Class

As you see the class has a private constructor and a public getInstance function that checks if the instance of the class is created or not. If yes, return it, else create a new one.