**Design Pattern:**

We have used three design patterns here.

* **Factory Pattern:**  We have used factory pattern to instantiate the different types of System users. Admin, Student and Professor all extends the user class. We create a UserFactoryClass to return the object of specific classes. This helps in creation of different objects through a shared common factory which helps in attaining a single point of control for multiple products.
* **Strategy Design Pattern**: We have used the strategy design pattern so that we can enable selecting an algorithm at runtime. As of now we will be implementing AST Strategy to check plagiarism. At later point of time we can implement different strategy and we can set the strategy and implement the strategy we want to in checkPlaigarism Interface which in turns executes the strategy which we want to implement.
* **Visitor Design Pattern:** We have used the visitor design pattern so that we can design reusable software. Here if we want to implement the functionality to build and AST for files in other languages we would just need to add another class and implement the visit method specific for building the AST for that class. The same has been represented in the UML class diagram.