Interfaces:

The login() method in Authentication interface allows Users to login into the system. The class implemented this interface will include implementation for the login process to the system. The forgotPassword() method in the interface Authentication is used to hide the process of sending the backup password and other implementation details as to how the user can again log into the system.

The User interface is implemented by the Instructors and Admin. Future additional users of the system can implement this Interface.

The interface Tree consists of a list of Nodes and these Nodes can be any type of Expression such as NumberExpression, StringExpression, VariableExpression etc., which will be implemented during the Implementation phase.

The interface Comparator is used to implement the plagiarism checks such as the functional breakdown, changing variable names, movingn the code around. Further plagiarism checking methods can be added to this interface and implemented by corresponding class.

Flow of class diagram:

The Instructor and the Admin will implement the User Interface. Both the types of Users, have e-mail, password, username and role which will allow them to login into the system via the Authentication Interface. The Authentication Interface implements the login and forgot password functionality.

The Admin has additional functionalities to create and delete accounts from the system. The Instructor can upload files, run the detector system and view the report of the plagiarism run.

The Uploader class has the github link as its member variable, which it will use to clone the repo on the system using its member functions.

The JSONSubmission returns a String of the submission files. In the Submission Class, it receives text representation of the code files and sets the student name and the homework number. This is a File object. The File class generates AbstractSyntaxTree which is represented using Node and Tree.

The PGRunner class receives list of AbstractSyntaxTrees and PGComparator class implements the functions to compare the AbstractSyntaxTree base on structure, variables, function body. The results from the comparison of the AbstractSyntaxTree are stored in the Report class members.

The PlagiarismDetector class implements the Singleton design pattern since we will have only one instance of the detector system.

The system implements Factory design pattern for Comparator.
The data structure used is HashMap with key as String(names of the two files) and value as list of two AbstractSyntaxTrees.

Since, our Detector System will clone from Github repositories, the sequence diagram is as follows: