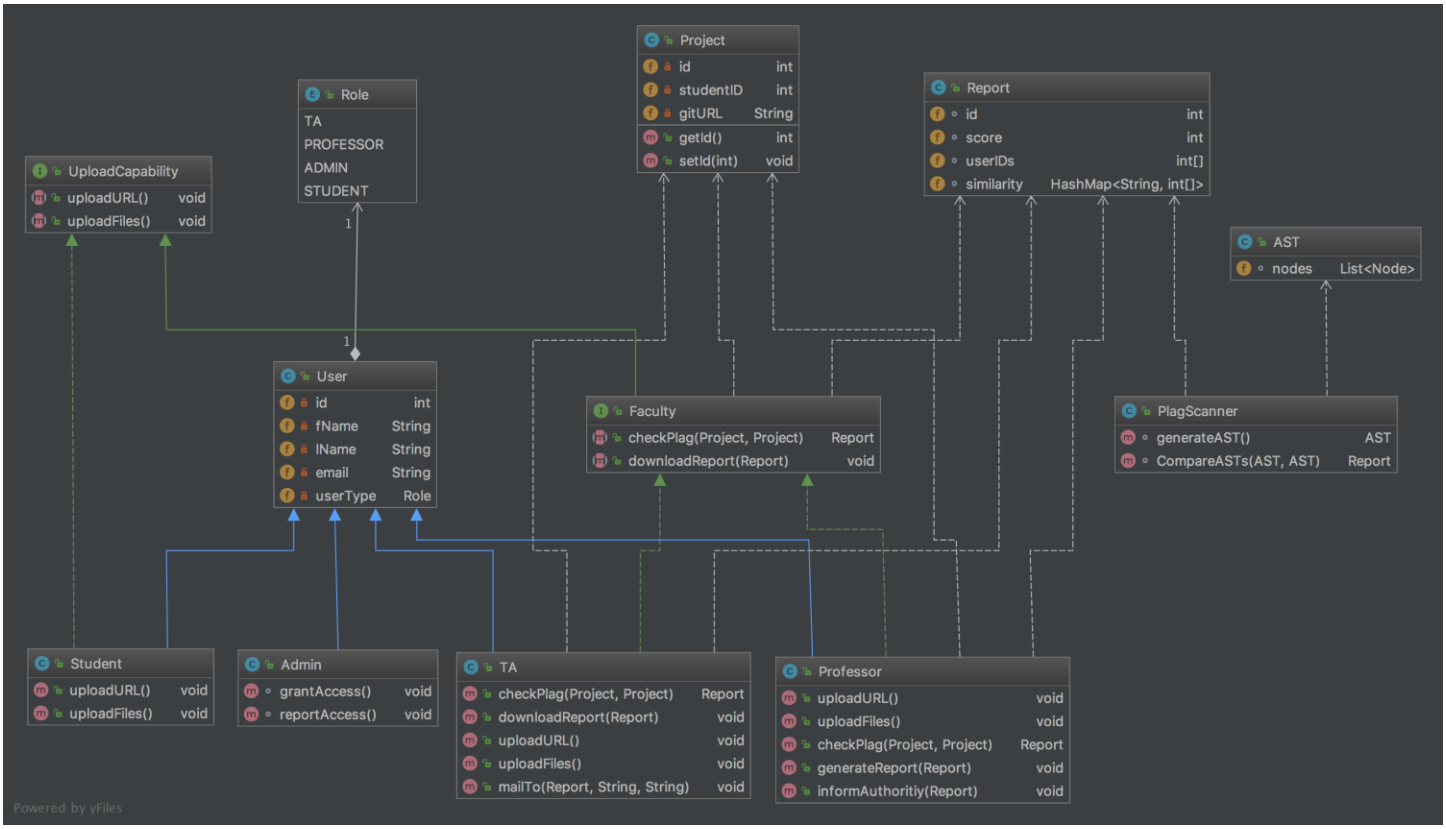
# CODE SNIFFER - PHASE B

## 1. UML CLASS DIAGRAMS:

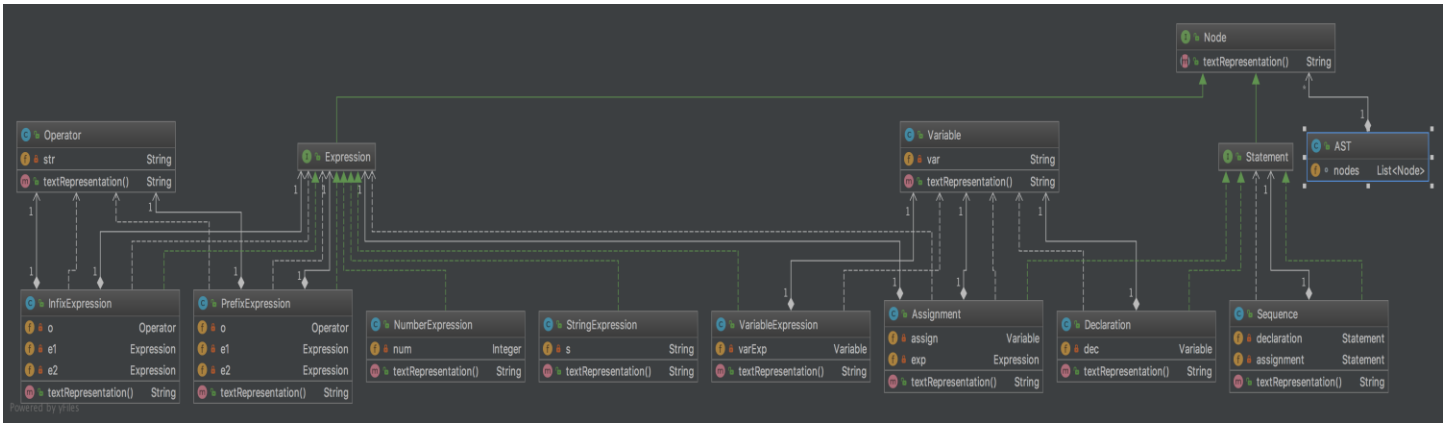
a. The class diagram describing the hierarchy of classes and interfaces that model the Code Sniffer project showing its attributes, methods and the relationship among the objects.

- User: It is a class representing the users of the system. There are only four allowed roles of the user namely, Professor, TA, Admin and Student represented by the classes Professor, TA, Admin and Student respectively.
- UploadCapability: It is an interface which provides features to upload the path of the location where project is present or upload the project files directly. It is implemented by the Student class and extended by the Faculty class.
- Faculty: It is an interface which differentiates between the users. TA and Professor are the faculty and only the faculty have the authority to check plagiarism and download the plagiarism report.
- Project: It is a class which represents the project. Each Project instance has a unique ID, Student ID and the git URL.
- Report: It is a class which represents the report obtained after two projects are found similar. Each report instance has a unique ID, the Plagiarism score (the extent up to which two files match), the IDs of students, and the similarity list (a map which has file name as key and the matching code lines as the value).
- AST: It is a class which represents the tree structure created after parsing the uploaded files.



b. The class diagram describing the hierarchy of classes and interfaces that model Abstract Syntax Trees data structure showing its attributes, methods and the relationship among the objects.

- AST: It is a class which represents the tree structure of a high-level code created after parsing the uploaded files. It consists a list of Node type.




## 2. INTERFACES AND DATA STRUCTURES:

The above class hierarchy has been implemented as the Java interfaces and classes along with the data structures. The proper documented code can be found in the folder “src” on GitHub.