

# **PLAGIARISM DETECTOR**

## **PHASE B IMPLEMENTATION**

### **Team 21**

Vishruth Krishna Prasad

Chaitanya Kaul

Ankita Patel

Rosy Parmar

## **Plagiarism Detector**

The main idea behind Plagiarism Detector is for the user to feed two .java files and check if the files are plagiarized.

Please refer phaseB-UML-Class-Diagram.png on the GitHub

## **FileType**

- The input to the Plagiarism Detector must be a .java file
- FileType is an enum

## **User**

- Login()- user logs in to the Plagiarism Detector by feeding the username, password

[After logging in, user is prompted to upload 2 files in the FileHandler]

## **FileHandler**

- Upload()- deals with uploading the input files from the local file system
- Parse()- gives the string representation of the whole code in the java file

[The user then starts the detector. The detector generates AST for the files and then returns the result]

## **Plagiarism Detector**

- fetchResults()- instantiates compareFactory class to return the results after comparing the 2 files
- generateAST()- feeds the input files to ASTFactory class to create an AST of the files

[The detector makes use of the two factories to create AST and compare the file/AST. The factories ensure that the AST creation and comparison of file/AST are hidden from the user]

## **ASTFactory**

- makeAST()- creates an AST for the given input files

[The file is converted to an AST and returns as per the respective function calls in the AST class]

## **AST**

The following methods are used to access and return a particular segment of the input file:

- getAllComments()
- getChildNodes()
- getMethodsInFile()
- getDataType()
- getBodyOfMethods()
- getClassBody()
- getCountOfLines()

## **Comparator**

- compareFiles()- implemented from the IComparator interface

## **IEnumerator**

- compareFiles()- a method with generalized input "T"

[CompareFactory compares the files/ASTs and returns the output]

## **CompareFactory**

The following methods are used to get the respective results from the classes that inherit from this class

- getLineCountAnalysis()
- getCompareCommentsAnalysis()
- getCompareTokensAnalysis()
- getCheckTransformationAnalysis()
- getCompareCodeRelocationAnalysis()

[Classes inherited from the **Comparator** Class are given below]

## **LineCount**

- compareFiles() [FileType is input]- implemented from the IEnumerator interface which is used to check if the 2 files have same line count

## **CompareComments**

- compareFiles() [AST is input]- implemented from the IEnumerator interface which is used to compare the comments from both the files

## **CompareTokens**

- compareFiles() [AST is input]- implemented from the IEnumerator interface which is used to compare the variables used in both the files

## **CheckTransformation**

- compareFiles() [AST is input]- implemented from the IEnumerator interface which is used to check if the 2 files have been refactored but contain the same implementation

## **CompareCodeRelocation**

- compareFiles() [AST is input]- implemented from the IEnumerator interface which is used to check if parts of code are moved around within the file of the 2 input files

## **DESIGN PATTERN:**

- We have used the Factory Method Design Pattern
- We have two factories namely, ASTFactory and CompareFactory which creates ASTs and Comparators respectively
- The Comparator implements IEnumerator and the methods from this interface is then overridden by its child classes
- The two factories are created such that the implementation of AST creation and Code/Tree comparisons are hidden from the client who uses the Plagiarism Detector
- Only the Plagiarism Detector is visible to the client