CODEX : Plagiarism Detection Tool

UserFactory
+makeProfessor(firstName, lastName, email, password)
+makeStudent(firstName, lastName, email, password)
+makeAdmin()

Student
+file: File
+firstName: String
+lastName: String
+email: String
+password: String
+contactNumber: Integer
+pathUrl: String
+files: List<File>
+registerUser()
+loginUser(email, password)
+logoutUser()
+uploadFile(pathUrl)

«Interface»
User
+loginUser(email, password)
+logoutUser()

File
+file_id: Integer
+fileName: String
+fileData: String
+timestamp: Date
+getFile(file_id): File

«interface»
ParserVisitable
+accept(ASTGeneratorVisitor)

«interface»
ASTGeneratorVisitor
+generateASTVisit(pythonParser)

PythonParser
+ast
+accept(ASTGeneratorVisitor): AST

As future scope, parsers for multiple languages can be implemented.

ConcreteASTGeneratorVisitor
+generateASTVisit(pythonParser): AST

ASTStrategy
+files: List<File>
+executeStrategy()
+generateAST(File): AST
+compareAST(AST ast1, AST ast2)
+generateReport()

Professor
+firstName: String
+lastName: String
+email: String
+password: String
+contactNumber: Integer
+students: List<Student>
+registerUser()
+loginUser(email, password)
+logoutUser()
+checkPlagiarism(Student, File)
+setStrategy()
+getStudents(): List <Student>

«interface»
CheckPlagiarismStrategy
+executeStrategy()

multiple strategies can be implemented for normalization and parsing in future scope.

«interface»
Comparator
+compare(AST ast1, AST ast2)

AST
+ast_id
+astList: List<String>

Report
+getPercentMatch()
+displaySummary()
+downloadReport()

Admin
+email: String
+password: String
+loginUser(email, password)
+logOutUser()
+updateUser()
+deleteUser()

ASTComparator
+asts: List<AST>
+compare(AST ast1, AST ast2)

+instantiates
1
1..*
+is a
+uploads
1
1..*
2..* +retrieves
+implements
1
+is a
+visitor : generateASTVisit(this)
1
1
+is a
1
1
1
1..*
+generates
1
1..*
+generates
+generates
1..*
+is a
+checks
1
1..*
+is a
+implements
1
+is a
+is a

Designed By : team-102