**UML**

### 1. Class Diagram

<<Java Interface>>
**User**
edu.neu.msd.plagiarismDetector
- inputEmail():String
- inputUserName():String
- inputPassword():String
- inputRole():String

<<Java Class>>
**Instructor**
edu.neu.msd.plagiarismDetector
- sub: List<Submission>
- rm: ReportManager
- pd: PlagiarismDetector
- emailId: String
- userName: String
- password: String
- role: String
- Instructor()
- login(String,String):boolean
- forgotPassword(String):void
- uploadSubmission(Submission):void
- deleteSubmission(Submission):void
- filterSubmission(Submission):void
- runPlagiarismDetector(List<Submission>):void
- cancelPlagiarismDetector():void
- viewReport(Report):void
- deleteReport(Report):void
- downloadReport(Report):void
- inputEmail():String
- inputUserName():String
- inputPassword():String
- inputRole():String

<<Java Class>>
**Admin**
edu.neu.msd.plagiarismDetector
- userId: String
- emailId: String
- userName: String
- password: String
- role: String
- Admin()
- login(String,String):boolean
- forgotPassword(String):void
- createAccounts(List<String>):boolean
- deleteAccount(List<String>):boolean
- inputUserName():String
- inputPassword():String
- inputRole():String
- inputEmail():String

<<Java Class>>
**PGRunner**
edu.neu.msd.plagiarismDetector
- mPairs: Map<String,List<AbstractSyntaxTree>>
- mReportManager: ReportManager
- mPGComparator: PGComparator
- PGRunner()
- createASTMap(List<Submission>):Map<String,List<AbstractSyntaxTree>>
- createReport():Report
- compare():void

<<Java Class>>
**PGComparator**
edu.neu.msd.plagiarismDetector
- report1: Report
- report2: Report
- report3: Report
- report : Report
- PGComparator()
- compareStructure(AbstractSyntaxTree,AbstractSyntaxTree):Report
- compareVariables(AbstractSyntaxTree,AbstractSyntaxTree):Report
- compareFunctionBody(AbstractSyntaxTree,AbstractSyntaxTree):Report
- combineReports(Report,Report,Report):Report

<<Java Class>>
**ReportManager**
edu.neu.msd.plagiarismDetector
- mReport: Report
- ReportManager()
- viewReport(Report):void
- generateReport():Report
- deleteReport(Report):void

<<Java Class>>
**PlagiarismDetector**
edu.neu.msd.plagiarismDetector
- mPlagiarismDetectorInstance: PlagiarismDetector
- mSubmissions: List<Submission>
- mReports: List<Report>
- getInstance():PlagiarismDetector
- PlagiarismDetector()
- runPlagiarismDetector(List<Submission>):boolean
- stopPlagiarismDetector():boolean

<<Java Interface>>
**Authentication**
edu.neu.msd.plagiarismDetector
- login(String,String):boolean
- forgotPassword(String):void

<<Java Class>>
**Uploader**
edu.neu.msd.plagiarismDetector
- gitLink: String
- submission: JSONSubmission
- subresult: List<Submission>
- Uploader()
- fetchSubmissionObject(String):JSONSubmission
- createSubmissionList(JSONSubmission):List<Submission>

<<Java Interface>>
**Comparator**
edu.neu.msd.plagiarismDetector
- compareStructure(AbstractSyntaxTree,AbstractSyntaxTree):Report
- compareVariables(AbstractSyntaxTree,AbstractSyntaxTree):Report
- compareFunctionBody(AbstractSyntaxTree,AbstractSyntaxTree):Report

<<Java Class>>
**Report**
edu.neu.msd.plagiarismDetector
- mCommonLiens: Map<Integer,Integer>
- mStructureMatch: String
- mTotalPercentMatch: int
- mFlag: String
- Report()
- getmCommonLiens():Map<Integer,Integer>
- setmCommonLiens(Map<Integer,Integer>):void
- getmStructureMatch():String
- setmStructureMatch(String):void
- getmTotalPercentMatch():int
- setmTotalPercentMatch(int):void
- getmFlag():String
- setmFlag(String):void

<<Java Class>>
**Submission**
edu.neu.msd.plagiarismDetector
- f: List<File>
- studentName: String
- hmkType: String
- Submission()
- getF():List<File>
- setF(List<File>):void
- getHmkType():String
- setHmkType(String):void

<<Java Class>>
**JSONSubmission**
edu.neu.msd.plagiarismDetector
- submission: String
- JSONSubmission()
- getSubmission():String
- setSubmission(String):void

<<Java Class>>
**File**
edu.neu.msd.plagiarismDetector
- timeSTamp: Date
- ast: AbstractSyntaxTree
- fileContents: StringBuilder
- File()
- createAST(StringBuilder):AbstractSyntaxTree

<<Java Class>>
**AbstractSyntaxTree**
edu.neu.msd.plagiarismDetector
- nd: List<Node>
- AbstractSyntaxTree()
- getNodes():List<Node>
- setNode(List<Node>):void

<<Java Class>>
**Node**
edu.neu.msd.plagiarismDetector
- value: String
- children: List<Node>
- Node()
- getValue():String
- setValue(String):void
- getChildren():List<Node>
- setChildren(List<Node>):void

<<Java Interface>>
**Tree**
edu.neu.msd.plagiarismDetector
- getNodes():List<Node>
- setNode(List<Node>):void
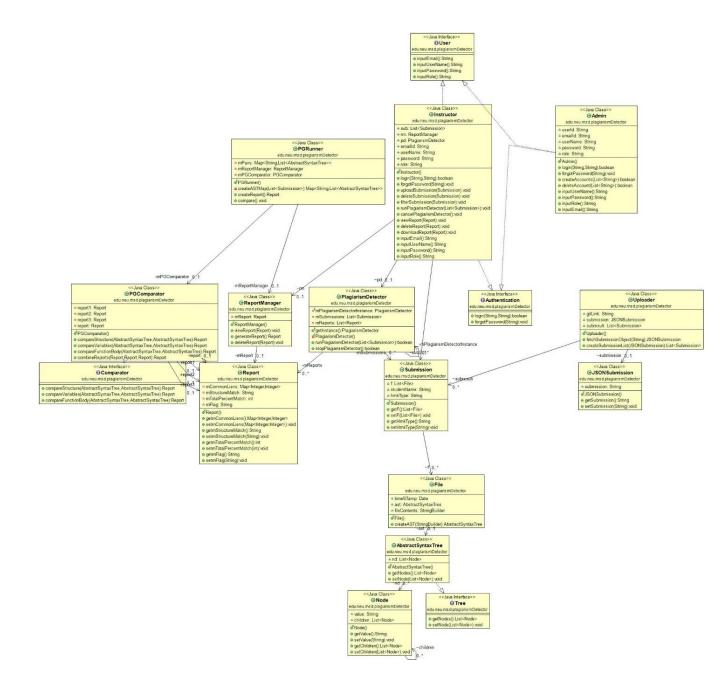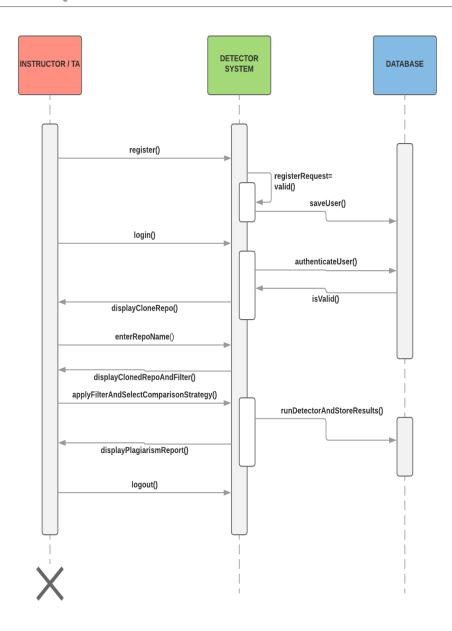
## 2. Sequence Diagram

## 3. Activity Diagram