Plagiarism Detector

George Dunnery, Jacob Piersall,
Yujia Liu, Akashbir Singh

Supported Languages

- Decision Process:
- Group familiarity was considered
- Share a similar syntax
- Both are Object Oriented
  - C++
  - Python

User Interface

- Choose two student projects
- Project can contain any number of files
- Select depth of comparison
- Press Detect to run
- View similarities in the side-by-side view

Plagiarism Detector

Select Project 1

Detect

Options

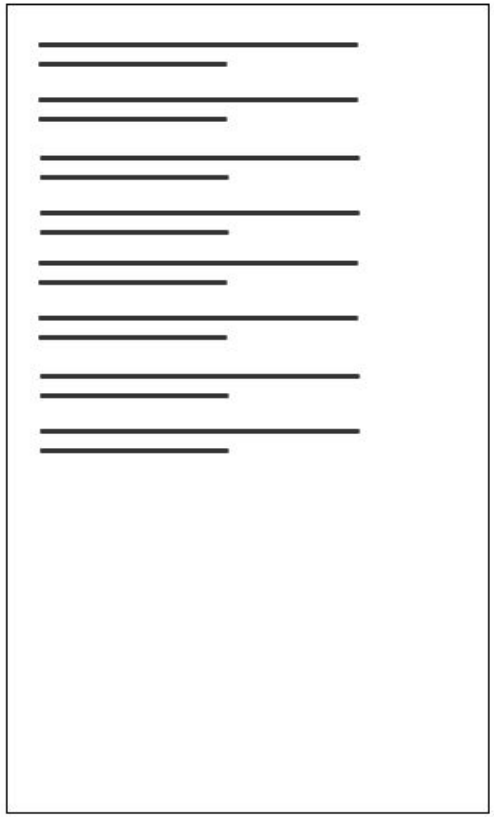
Similarity

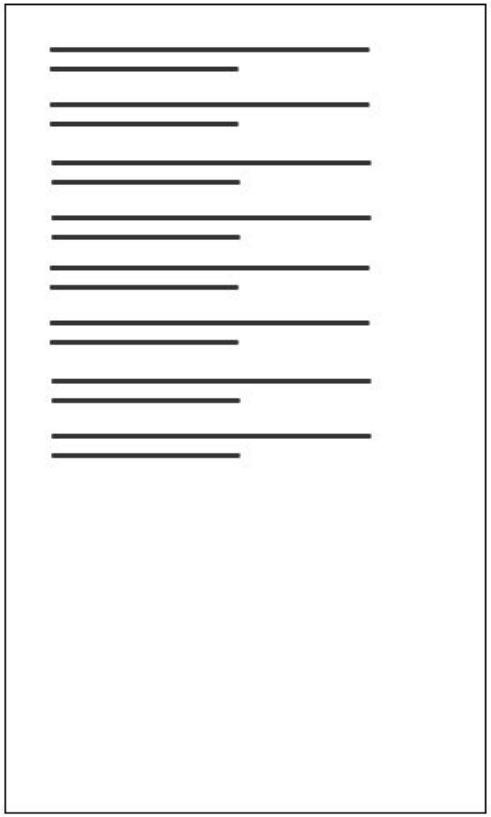
Select Project 2

File.py
31 KB

File.py
37 KB

File.py
102 KB

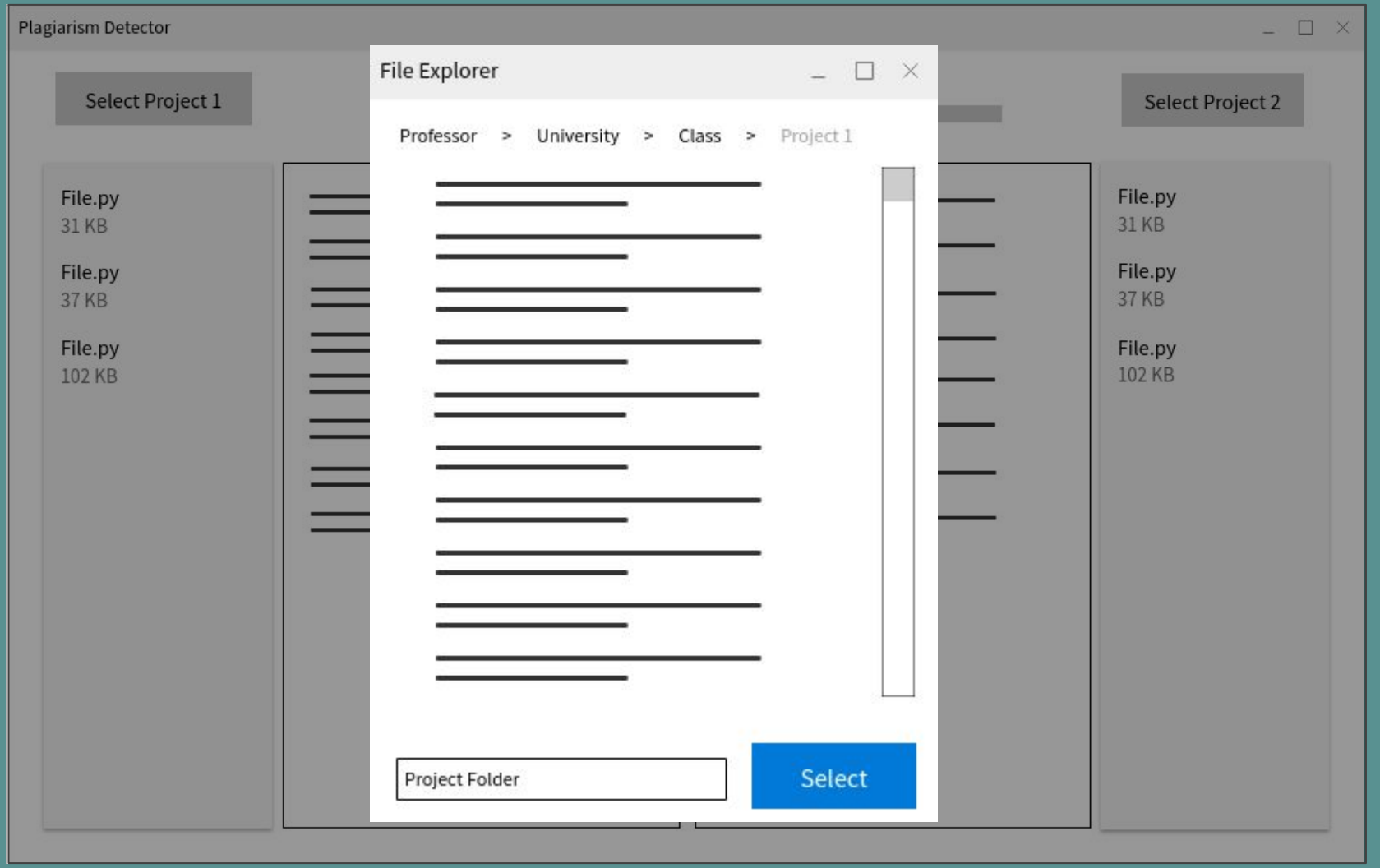


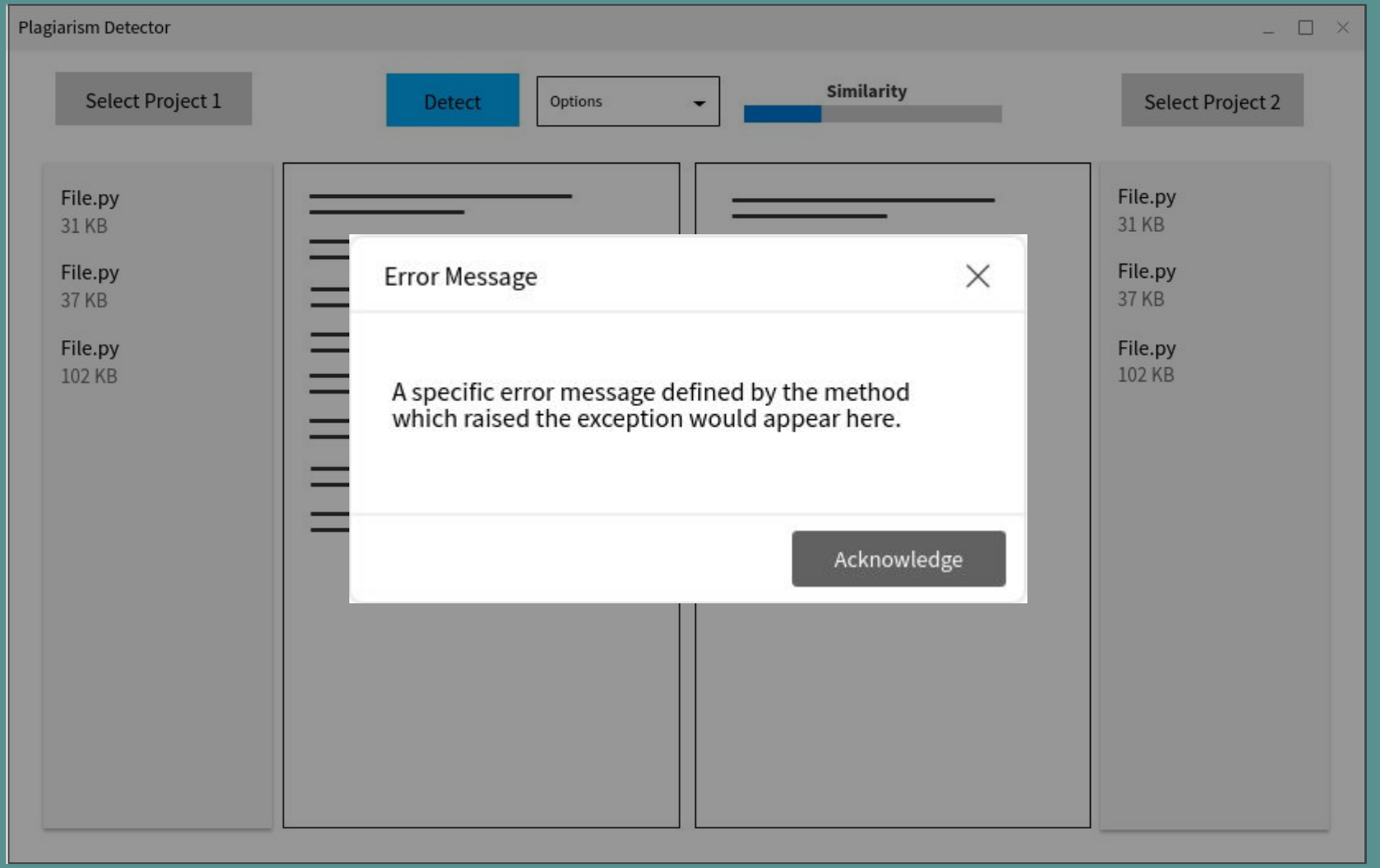


File.py
31 KB

File.py
37 KB

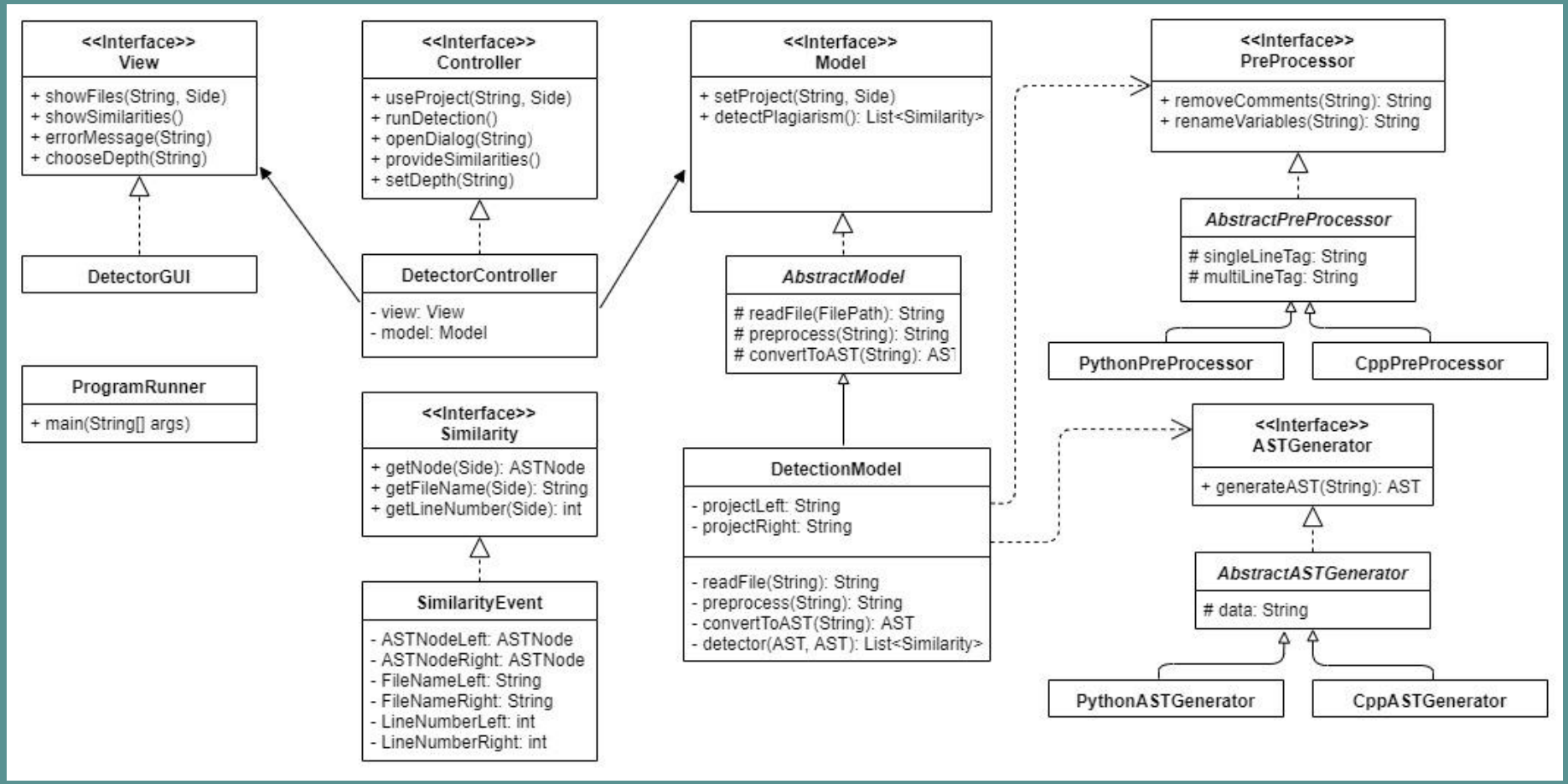
File.py
102 KB

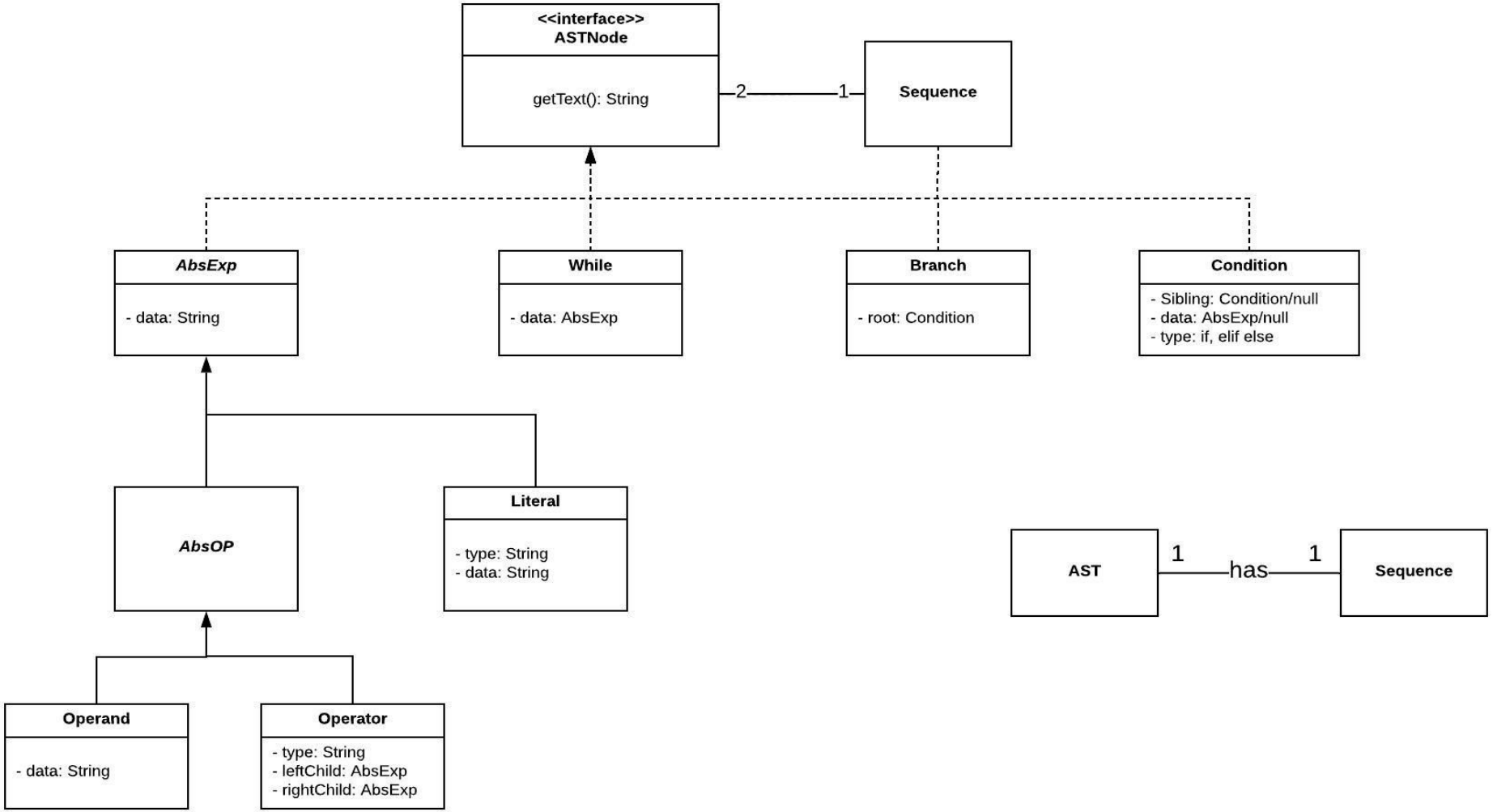

Plagiarism Detector
Select Project 1
File.py
31 KB
File.py
37 KB
File.py
102 KB
File Explorer
Professor > University > Class > Project 1
Project Folder
Select
Select Project 2
File.py
31 KB
File.py
37 KB
File.py
102 KB

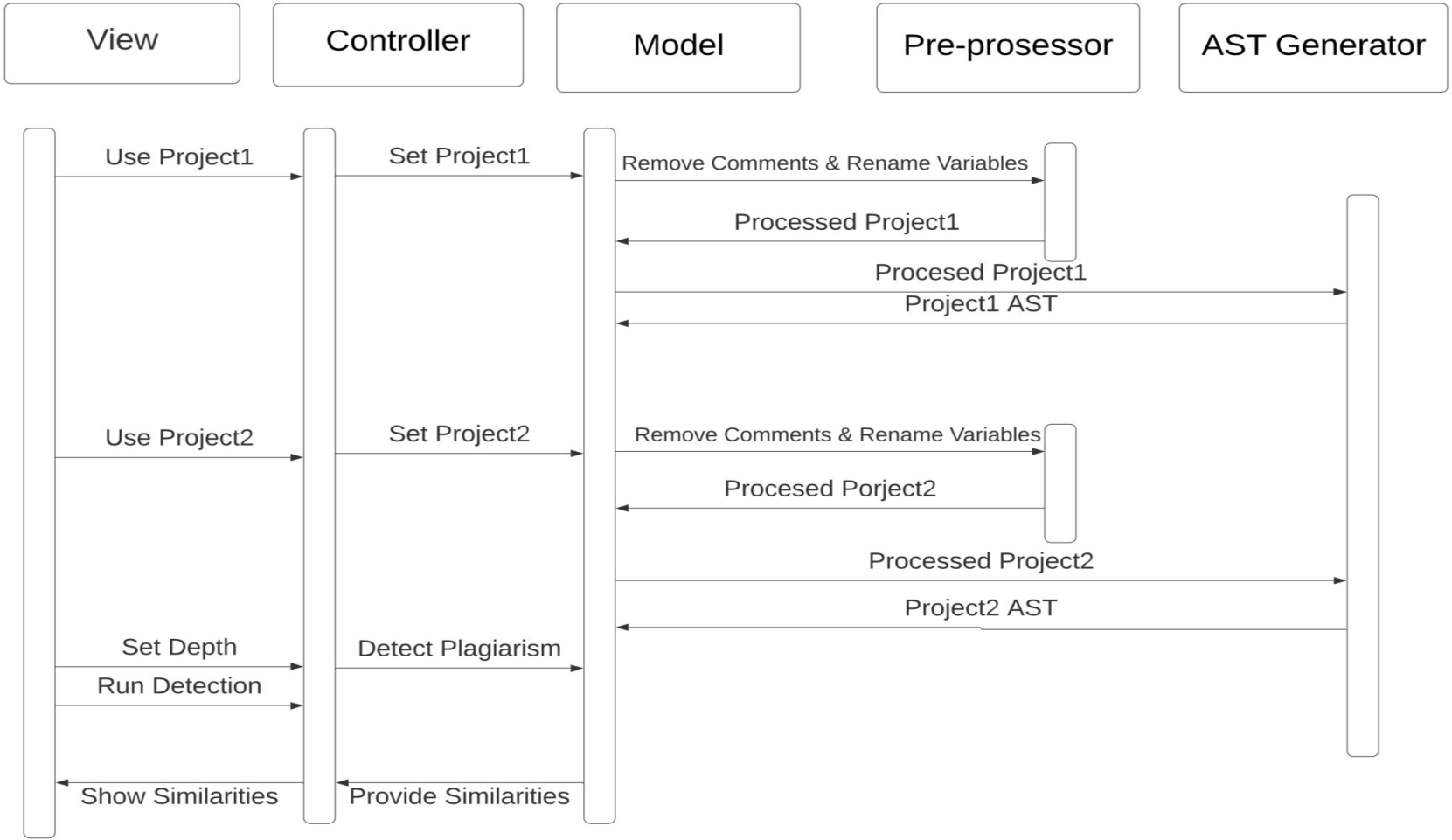

Plagiarism Detector
Select Project 1
Detect
Options
Similarity
Select Project 2
File.py
31 KB
File.py
37 KB
File.py
102 KB
Error Message
A specific error message defined by the method which raised the exception would appear here.
Acknowledge
File.py
31 KB
File.py
37 KB
File.py
102 KB

UML Diagrams

- Class diagram
- AST Data Structure Diagram
- Sequence Diagram


<<Interface>>
View
+ showFiles(String, Side)
+ showSimilarities()
+ errorMessage(String)
+ chooseDepth(String)
DetectorGUI
ProgramRunner
+ main(String[] args)
<<Interface>>
Controller
+ useProject(String, Side)
+ runDetection()
+ openDialog(String)
+ provideSimilarities()
+ setDepth(String)
DetectorController
- view: View
- model: Model
<<Interface>>
Similarity
+ getNode(Side): ASTNode
+ getFileName(Side): String
+ getLineNumber(Side): int
SimilarityEvent
- ASTNodeLeft: ASTNode
- ASTNodeRight: ASTNode
- FileNameLeft: String
- FileNameRight: String
- LineNumberLeft: int
- LineNumberRight: int
<<Interface>>
Model
+ setProject(String, Side)
+ detectPlagiarism(): List<Similarity>
AbstractModel
readFile(FilePath): String
preprocess(String): String
convertToAST(String): AST
DetectionModel
- projectLeft: String
- projectRight: String
- readFile(String): String
- preprocess(String): String
- convertToAST(String): AST
- detector(AST, AST): List<Similarity>
<<Interface>>
PreProcessor
+ removeComments(String): String
+ renameVariables(String): String
AbstractPreProcessor
singleLineTag: String
multiLineTag: String
PythonPreProcessor
CppPreProcessor
<<Interface>>
ASTGenerator
+ generateAST(String): AST
AbstractASTGenerator
data: String
PythonASTGenerator
CppASTGenerator


<<interface>>
ASTNode
getText(): String
2
1
Sequence
AbsExp
- data: String
While
- data: AbsExp
Branch
- root: Condition
Condition
- Sibling: Condition/null
- data: AbsExp/null
- type: if, elif else
AbsOP
Literal
- type: String
- data: String
AST
1
has
1
Sequence
Operand
- data: String
Operator
- type: String
- leftChild: AbsExp
- rightChild: AbsExp


View
Controller
Model
Pre-prosessor
AST Generator
Use Project1
Set Project1
Remove Comments & Rename Variables
Processed Project1
Procesed Project1
Project1 AST
Use Project2
Set Project2
Remove Comments & Rename Variables
Procesed Porject2
Processed Project2
Project2 AST
Set Depth
Detect Plagiarism
Run Detection
Show Similarities
Provide Similarities

Pre-Processing

- Removal of comments
- Renaming variables in the order they are encountered
- Combining project files into one large file
- Removal of other Lexical Modifications

Algorithms

- Longest Common Subsequence
  - Only useful for directly copied code
  - Well documented in CLRS
  - Relatively easy to implement as a shallow analysis

Algorithms

- Sub-tree clone detection using Abstract Syntax Tree
- Research paper describes an algorithm for large scale to AST clone detection
- Source: http://leodemoura.github.io/files/ICSM98.pdf

Potential Improvements

- Professor Login
- Web submission portal for students
- Support for different courses
- SQL Database to store & retrieve project files

The End

- Questions?