

Plagiarism Detector

George Dunnery, Jacob Piersall,
Yujia Liu, Akashbir Singh





Supported Languages

- C
- C++
- Python
- and more!!!*

*Partial support exists for ALL languages and files. (Disabled in this version)

*Full support for additional languages can be added in about 10-15 minutes each



User Interface

- Choose two student projects
- Project can contain any number of files
- Select algorithms to use in comparison
- Press Detect to run
- Highlight similarities in the side-by-side view
- Programming language use is auto-detected

Select Project 1

Detect

Options



Similarity



Select Project 2

File.py

31 KB

File.py

37 KB

File.py

102 KB

File.py

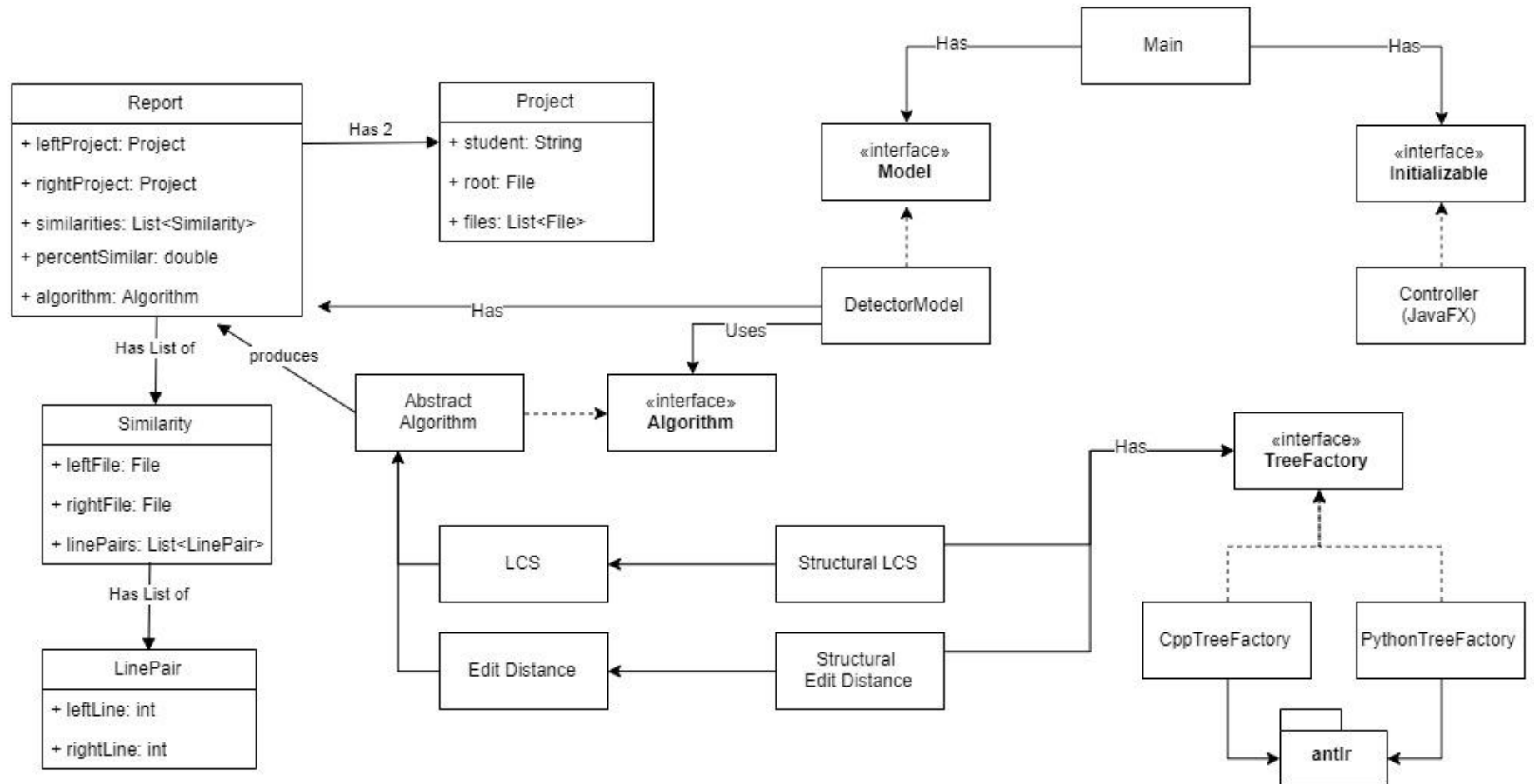
31 KB

File.py

37 KB

File.py

102 KB





Content Comparisons

- Compare contents of two projects
- Algorithms
 - Longest Common Subsequence
 - Edit Distance



Antlr Structural Comparisons

- Structural versions of LCS and Edit Distance used to compare code structure instead of contents
- The result of this transformation are passed into the LCS and Edit Distance algorithms
- Before: `int i = 0; j = i + 42;`
- After: `type name = num; name = name + num;`



Notes for (potential) Improvements

- Improve visual layout of the View
- Optimizing code base with refactoring
- Algorithm optimization for threading
- Addition of tree-based algorithms
- Use a visitor as part of an algorithm
- Quality of Life Improvements and Minor Bugs



Things we Learned and would do Differently in the Future

- Improvements to separation of tasks earlier to better assist in division of labor
- Better planning of required classes near the start
- Following SCRUM closer to ensure steady development throughout the work period



The End

- Questions?