Project: plagiarism-detector  
Specifications

Olivier Duménil  
Antoine Jacquin-Ravot

Table of Contents

[I/ Scope of the Project 3](#_Toc390949776)

[II/ Description of the project 3](#_Toc390949777)

[a/ The core applications 3](#_Toc390949778)

[b/ The detection modules 3](#_Toc390949779)

[c/ Interfacing the core and the modules 3](#_Toc390949780)

# I/ Scope of the Project

This project aim to produce a software able to help teachers detect plagiarism in programming assignments. It should provide the following features: detection methods modularity, ease of access, results readability.

The creation of new detections algorithms is out of scope.

# II/ Technical Specifications

## A/ Core application

* 1. Menu Bar
     1. File
        1. Exit: exit the program
        2. Export: open the “Export” window
     2. Edit
        1. Cut
        2. Copy
        3. Past
        4. Undo
        5. Redo
     3. Modules
        1. Manage modules: open the “Manage Modules” window
        2. Select module(s): open the “Select Modules” window
     4. Sources
        1. Add sources folder(s): open a directory browser
        2. Add sources file(s): open a file browser
        3. Manage sources: open the “Manage Sources” window
        4. Select sources skeleton: open the “Select Sources Skeleton” window
     5. Analysis
        1. Start/Resume: start/resume the analysis
        2. Stop: cancel the analysis
        3. Pause: pause the analysis
     6. Help
        1. About: open the “About” window
        2. Documentation: open the “Documentation” window
  2. Configuration view
     1. Module Selector
        1. Name of each module
        2. Checkbox to activate/deactivate the module
        3. Module configuration button: open the configuration form provided by the module in the “Module Configuration” window
     2. Sources Browser
        1. Sources drag’n’drop area
        2. Add sources file(s) button: open a file browser
        3. Add sources folder(s) button: open a directory browser
        4. Delete button: delete the selected file(s) and/or folder(s)
     3. Analysis Status
        1. Start/resume button
        2. Stop button
        3. Pause button
        4. Progress bar
        5. Status label
  3. Pop-up windows
     1. Export
        1. Export format picker
        2. Export button
        3. Cancel button
     2. Manage Modules
        1. List of currently loaded modules
        2. Checkbox to activate/deactivate the module
        3. Module configuration button: open the configuration form provided by the module in the “Module Configuration” window
        4. Add module button: open a file browser and load the selected file(s)
        5. Delete module button: unload the selected module(s)
     3. Manage Sources
        1. Sources drag’n’drop area
        2. Add sources file(s) button: open a file browser
        3. Add sources folder(s) button: open a directory browser
        4. Delete button: delete the selected file(s) and/or folder(s)
     4. Select Sources Skeleton
        1. Sources Skeleton radio list
        2. Add sources skeleton button: open a file browser
        3. Delete sources skeleton button: delete the selected file(s)/folder(s)
     5. About
        1. Display information about the software, its version, the project
     6. Documentation
        1. Display the documentation file
     7. Module Configuration
        1. Display the configuration form provided by the module
        2. Validate button
  4. Result view
     1. Display the results

## B/ Application Programming Interface

The detection modules must implement the following methods to be used by the core application

1. Module API
   1. GetModuleInformation: Send the name of the module, its version and miscellaneous information about the module
   2. GetParameterForm: Send the form to be used to configure the module
   3. SetParameters: set the parameters to be used by the module
   4. Start: begin the analysis
   5. Resume: resume the analysis
   6. Pause: pause the analysis
   7. Stop: cancel the analysis
   8. SetSources: set the folders and files to be analysed
   9. GetResults: send the results of the analysis when availables
   10. SetDelegate: set the delegate to be used to notify the core application
2. Delegate API
   1. SetStatus: set the status of the analysis
   2. SetProgress: set the progress of the analysis