# 5. BONUS FEATURES

Our Static Program Analyser (SPA) has been extended beyond its call of duty. The extended (extra) features will be explained in more details below:

## FLEXIBLE CODEPARSER

The source code tested for the SPA, is defined to be in standardized format and neatly arranged. They have regular and consistent spacing, indentation, tabs, and endline characters. Please refer to the example below:

procedure Orchid {

while idx {

y = z\*3 + 2\*x;

call Tulip;

idx = idx - 1; }

z = z + y + idx; }

procedure Lily {

while v {

y = x + y; }

x = y + x; }

However, our SPA source code parser is very flexible. It is able to detect erroneous and inconsistent spaces, tabs, endline characters and erroneous close curly brackets. A flexible CodeParser is one of our extended features, as it is not required to build one for this project. To demonstrate the ability of our CodeParser, please refer to the example below (which will have the same information/result as the example above):

procedure Orchid

{

while idx {

y = z \*3 + 2\* x ;

call Tulip ;

idx = idx -1;

}

z=z+y+idx;

}

procedure Lily{

while v{

y =x + y ;}

x = y + x;}

## FLEXIBLE QUERYPARSER

The queries being tested with SPA in this module are defined to be in standard format. They have regular and consistent lowercase/uppercase command, spacing, and characters. Please refer to the sample standard query below:

if ifstat; Select ifstat such that Follows\* (ifstat, 17)

assign a; Select a such that Modifies (a, "idx") and Uses (a, "idx")

assign a; while w; Select a such that Modifies (a, "idx") and Uses (a, "idx") and Follows (15, a) and Parent\* (w, a)

However, our SPA QueryParser is very flexible and able to detect queries with inconsistent/extra spaces and lowercase/uppercase clauses. It is not required to build a flexible QueryParser for this project. To demonstrate the ability of our QueryParser, please refer to the example below (which will have the same information/result as the example above):

If ifstat ; Select ifstat such that Follows\* ( ifstat,17)

assign a;Select a such that Modifies(a,"idx") and Uses(a,"idx")

assign a ; while w ; Select a such that Modifies (a, "idx" ) and Uses ( a , "idx") and Follows (15, a) and Parent\* ( w , a )

## HIGHLY ORGANIZED REPOSITORY

Our codes and files in the repository are very organized. We follow standard team naming conventions and structure the directory in the repository to make sure every team member understands and has easy access.



Appropriate milestones and issues are also tracked and reported regularly to achieve any goals or/and objectives.





Every commit related to certain issues or milestones will be tagged and linked to the issue for easy tracking and reference in the future. It is also used to monitor the progress of any issues and the SPA system as a whole. Github features such as milestones, issue tracker with assignees, milestones, and issue category (bugs, testing, documentation, enhancement, etc) are highly utilized by our team.





Our API and code is also well-documented. Every detail of the code is described clearly, showing and explaining all methods, attributes, and inheritance diagram whenever applicable.



In conclusion, our progress in building this project is thoroughly tracked and documented so that anyone in the future will be able to understand project with ease.

Our project is hosted on GitHub, and the website links are as follows:

* Repository : <https://github.com/yulonglong/Static-Code-Analyzer>
* Issue tracker : <https://github.com/yulonglong/Static-Code-Analyzer/issues>
* Milestones : <https://github.com/yulonglong/Static-Code-Analyzer/milestones>
* Commits : <https://github.com/yulonglong/Static-Code-Analyzer/commits/master>