SCAPES

Functional Model Document

By:

Redwan Wadud

100873111

Submitted to:

Dr. Christine Laurendeau

COMP 3004 - Object-Oriented Software Engineering

School of Computer Science

Carleton University

October 1, 2019

1. Functional Requirements

F01 Programmers/System Administrators can write a program in the SCAPL language as per the SCAPL instruction set.

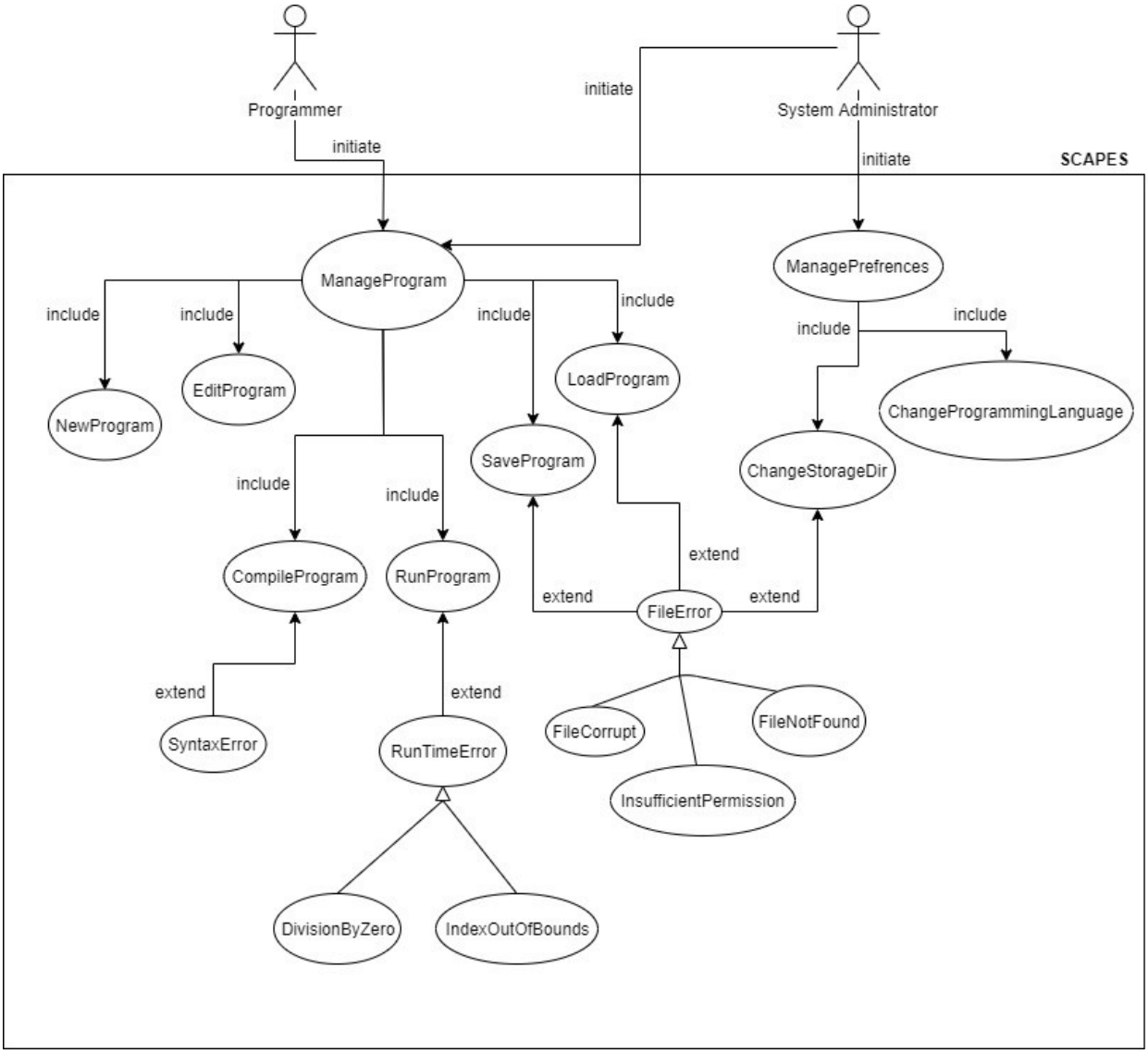
F01.01 Programmers/System Administrators can write a new program.
F01.02 Programmers/System Administrators can edit an existing program.
F02 Programmers/System Administrators can save a program to a file.
F03 Programmers/System Administrators can load a program from a file.
F04 Programmers/System Administrators can compile an existing program.
F05 Programmers/System Administrators can run an existing compiled program.
F06 System Administrators can manage system preferences.
F06.01 System Administrators can change preferred programming language.
F06.02 System Administrators can change storage directories.

2. Non Functional Requirements

Usability	
NF01	The system should verify the syntax of the program and prompts/highlights any syntax errors made to the user.
NF02	An option should be present to allow the user to compile and run a program in a single click.
NF03	The system should allow the user to select a color scheme such as light mode/dark mode.
Reliability	
NF04	The system should autosave the program after every line entry or whenever the SCAPES editor goes out of focus.
NF05	The system should store copes of previous revisions of the program, in case the user wants to compare and or restore to an older version of the code.
Performance	
NF06	Compilation should not take more than 1 second.
NF07	The program editor should not be laggy regardless of the number of lines of code.
Supportability	
NF08	The system should be supportable in any OS environment.
NF09	The system should provide automatic updates if allowed by the user.
Implementation	
NF10	The system must be written in c++ using the QT framework.
NF11	The system should be designed in such a way that it can be implemented in any object oriented language with ease.
Interface	
NF12	Any communication with an external system should be encrypted.
NF13	The system should have the ability to integrate with git.
NF14	The system should have the ability to support any external cloud storage systems (Google Drive, Onedrive, …etc)
Operations	
NF15	The system software should be able to be installed and built with only 1 command.
NF16	The user should be able to run a program with only 1 command.
NF17	The development team will provide 1 year of support to the client from the date of purchase.
Packaging	
NF18	The system should be packaged as a single executable file.
Legal	
NF19	Proper credits should be given to any third party libraries used, in order to avoid copyright issues.

3. Use cases

3.1 Full use case diagram of entire SCAPES system.



3.2 Use case table descriptions

Use case Id	UC-01
Name	**ManageProgram**
Participating actors	Initiated by Programmer/System Administrator
Flow of events	1. The user can choose to write a new program, load an existing program, compile a program, run a program, or save a program as handled by the included use cases **NewProgram, SaveProgram, LoadProgram, CompileProgram, RunProgram.**
Entry conditions	
Exit conditions	The user exits the program.
Quality requirements	
Traceability	F01.01, F01.02, F02, F03, F04, F05

Use case Id	UC-06
Name	**CompileProgram**
Participating actors	Initiated by Programmer/System Administrator
Flow of events	1. The user selects the option to compile the program 2. The system verifies the syntax of the program 3. If there are any syntax errors, the compilation is aborted. 4. The system then notifies the user of any syntax errors. 5. If no errors are detected, the system proceeds to translate the program. 2. The user is prompted to run the program after successful compilation.
Entry conditions	
Exit conditions	Program successfully is compiled or aborted due to an error
Quality requirements	Compilation should take no more than a second. Detected errors should be highlighted in the editor.
Traceability	F06, NF01

Use case Id	UC-07
Name	**RunProgram**
Participating actors	Initiated by Programmer/System Administrator
Flow of events	1. The user selects to run an already existing compiled program. 2. The user selects to compile & run a new program.
Entry conditions	The program must have been compiled.
Exit conditions	The program terminates.
Quality requirements	Any runtime errors should be notified to the user.
Traceability	F05

Use case Id	UC-10
Name	**SyntaxError**
Participating actors	Programmer/System Administrator
Flow of events	1. The system encounters a syntax error while verifying the syntax of the program. 2. The system continues to scan the program for additional syntax errors. 3. The system highlights the syntax errors and notifies the user.
Entry conditions	A syntax error is detected
Exit conditions	The compilation is aborted
Quality requirements	
Traceability	NF01