SOFTWARE REQUIREMENTS SPECIFICATION FOR "RESHARPER – CODE REFACTORING TOOL"

Version 1.3

Prepared By
SHAMS BIN SOHRAB

AUGUST 23, 2017

AMERICAN INTERNATIONAL UNIVERSITY BANGLADESH

Contents

Revision History

Name	Date	Reason For Changes	Version
Shams Bin Sohrab	9/07/17	Initial release	1.0
Shams Bin Sohrab	13/07/17	Change in section 2.6 User Documentation	1.1
Shams Bin Sohrab	23/08/17	Changes in section 4 System Features	1.2

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the ReShaper - Code Refactoring Tool. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

1.2 Document Conventions

This document follows MLA Format. Bold-faced text with font size 18 has been used to emphasize section and font size 14 for sub-section headings. Highlighting is to point out words in the glossary and italicized text is used to label and recognize diagrams.

1.3 Intended Audience and Reading Suggestions

This document is written for the researchers, software developers, documentation writers, and users involved as the part of beta testing. In the next section, system features with their functional requirements are presented to highlight the major services provided by the intended product. Then the external interface requirements highlighting the logical characteristics of each interface between the software product and the users are discussed. Finally, this specification is concluded with the reference documents on which this document is based on.

1.4 Product Scope

ReShaper is a programmer friendly tool for the most convenient way of programming. It enables efficiency and reduce redundancy. The touch of intelligence added in this program is definitely a big help for the users as it can point common errors and suggest possible solution by itself. Functionalities like automated annotations will increase source code’s readability. The best part is this facility will be provided for most commonly used language in this same platform. It can used both as an editor or an extension to other tools like Visual Studio, NetBeans and Eclipse etc.

1.5 References

1) IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.
2) Source Code Analytics with Roslyn and JavaScript Data Visualization by Sudipta Mukherjee.
3) Software Requirements Specification - Amazing Lunch Indicator http://www.cse.chalmers.se/~feldt/courses/reqeng/examples/srs_example_2010_group2.pdf

2. Overall Description

2.1 Product Perspective

The final product will be an intelligent code refactoring tool that can improve the functional and non-functional attributes mostly by the program's increased readability and reduced complexity. Though most IDE offer this feature but they offer for one particular language. But this product is going to provide this support for all commonly used programming languages. C++, JAVA and C# are the languages that will be supported by this application in the initial release. A programmer, as the user of the product just need to open the editor and write source code. It will keep track of the variables user used, functions that are created or added from library and follow common programming convention for that particular language automatically. Suggestion of keywords and the variables will be a helpful and make the process of writing source code faster. The features in terms of testing will make it easier to test an unfinished program.

The product can be used as an editor or an extension to other products Like Visual Studio, NetBeans and Eclipse.

2.2 Product Functions

ReShaper- Code Refactoring Tool will provide following main features-

1) **Maintaining proper notations:** The product will automatically follow or suggest common practices and conventions like Camel notations, Pascal notations, Author annotation etc.
2) **Automated stub for testing:** The program will take users command and automatically create working stubs for testing and debugging purpose.
3) **Removal of redundant functions and variables:** If any variable or functions are declared but not used will be removed automatically.
4) **Suggestions about alternative options:** A programmer can choose from various functions any language provides from the suggestion bar. This process will be autonomous and will cover both user defined and language provided functions, properties and variables.
5) **Dispensing comments and duplicate codes:** If user wants it will dispense any unnecessary parts such as comments, unused libraries from the code. If any duplicate code is found the program will dispense that portion immediately.

2.3 User Classes and Characteristics

The user of this product will be professional programmers and software developers as well as testers. They must have proper knowledge about the language they are willing to use. Also knowledge about basic programming is a must. If the user intends to use the program as an extension of other development tool than proper knowledge about that parent program is also essential. Students who are learning to program can use this product too. They will need proper understanding about basic programming and use of their desired programming language.

2.4 Operating Environment

In the initial release the program will only work for Windows operating system. All edition of Windows 7, Windows 8, Windows 8.1 and Windows 10 are supported by this programming. The program can be used as an extension. In that case the parent program must be installed in supported Windows editions too.

2.5 Design and Implementation Constraints

1) The product must support all editions of Windows 7 and higher.
2) The program must be able to run under several other development tools. For the initial release it must be able to run under Visual Studio 2012, Visual Studio 2015, NetBeans and Eclipse
3) The product should be able to read commonly used programming languages. For the initial release C++, JAVA and C# must be supported by the program.
4) It should not take more than 700 megabytes of memory.

2.6 User Documentation

Along with the software product, a user manual would be written to help users understand the working methodology. It would be written for nontechnical individuals such as students and the level of content or terminology would differ considerably this document which is more detailed and complex. The user manual would follow common user documentation styles capturing purpose and scope of the product along with key features and operations, step-by-step instructions for using the program including conventions, tips for errors and malfunctions and glossary of terms.

2.7 Assumptions and Dependencies

This product is dependent on any update from any programming language. For any update of a programming language an update of the product must be released. Also as an extensible program to other development tools the user must be using those tools with administrative privileges. Any kind of changes in parent program will also make an update of the program imminent.

3. External Interface Requirements

This section will discuss the working process of "ReShaper – Code Refactoring Tool" via a block diagram.

3.1 Block Diagram

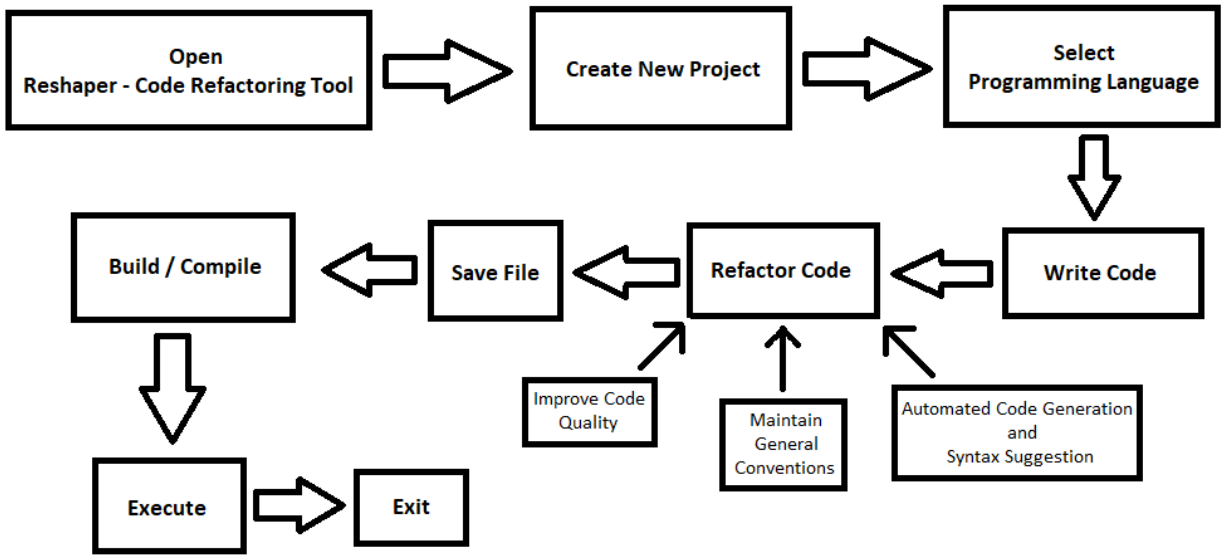


Fig: Block Diagram of ReShaper – Code Refactoring Tool

The user will start by opening the software "ReShaper - Code Refactoring Tool". After the software has been started the user has to create a new project by entering the project name and select the programming language he wants to develop in. The system will automatically include some commonly used library files. So, the user should not bother about manually including those library files to the project. The user can start writing his code. The system will provide him some flexibility by automatically writing some repeated codes and while coding the user will also enjoy real-time syntax suggestions for the programming language he selected. The refactoring functionality of the system will help the user to improve his code quality. It will also warn the user to follow proper programming language conventions which will help the code to be more manageable and readable.

3.2 Software Interfaces

The system can be used as a downloadable plugin available for Microsoft Visual Studio, NetBeans and Eclipse.

4. System Features

This section will provide an overview of all the available features of "ReShaper – Code Refactoring Tool"

4.1 System Feature 1

Title: Create / Open File.

Priority: High.

Description: This feature will allow the user to create new project files as well as open previously written files. For creating new project file the user must provide a name of the file and the preferred programming language he wants to develop his project in. A new project file will be created with the extension of the programming language that the user has selected. The system will automatically include most commonly used library files.

Functional Requirement: REQ-1

Dependencies: None.

4.2 System Feature 2

Title: Write / Edit Code.

Priority: High.

Description: The user will be provided with a friendly environment for writing and managing source codes. The system will ease the process of coding via providing many useful functionalities.

Functional Requirement: REQ-2

Dependencies: REQ-1

4.3 System Feature 3

Title: Save File.

Priority: High.

Description: The user should be able to save the file by clicking File **->** Save or by using the keyboard shortcut (ctrl) + (s).

Functional Requirement: REQ-3

Dependencies: None.

4.4 System Feature 4

Title: Build / Run.

Priority: High.

Description: The system will have integrated compilers for C, C++, JAVA, C#, PHP and Python. It will provide the functionality to compile user's code as like other IDE's. This compilation process will detect language based syntax, logical and operational errors.

Functional Requirement: REQ-4

Dependencies: REQ-2

4.5 System Feature 5

Title: Refactoring Code.

Priority: High.

Description: This is the unique feature of the system. The system will not only detect compilation errors but also some logical, notational, conventional and quality related errors prior to compilation of the code. The system strictly follows notational and conventional rules for some basic programming languages like C++, JAVA and C#. It will provide warning messages to the user if the user does not follow proper conventional and notational rules. The user will be suggested to fix those warning messages before the compilation of the source code.

Functional Requirement: REQ-5

Dependencies: REQ-2

4.5.1 System Feature 5.1

Title: Logical Error Tracing.

Priority: Medium.

Description: This tracing functionality will basically trace if there is any redundant code segments in the code, again it will also detect if any part of the code is unreachable due to logical statements provided by the user. If there is any chance that the code may fall into infinite loop then the system will also provide a warning message to the user.

Functional Requirements: REQ-5.1

Dependencies: REQ-2

4.5.2 System Feature 5.2

Title: Basic Quality Testing.

Priority: Medium.

Description: The system will detect possible patterns of errors and faults that may lead to failure of the users program. Primarily, it will detect data flow and control flow anomalies and provide warning messages to the user.

Functional Requirement: REQ-5.2

Dependencies: REQ-2, REQ-5.1

4.5.3 System Feature 5.3

Title: Follow Proper Convention and Notation.

Priority: High.

Description: The system will abide by the conventional and notational rules of programming languages. It will provide warning messages to the user if the user has not followed proper convention of that particular programming language. In first release the system will have general notational and conventional rules for basic programming languages like C++, JAVA and C#. Following proper conventions and notations results in more readable and maintainable code.

Functional Requirement: REQ-5.3

Dependencies: REQ-2

4.6 System Feature 6

Title: Real-time Syntax Suggestions.

Priority: Medium.

Description: The system will provide automated suggestions for programming languages from included libraries.

Functional Requirement: REQ-6

Dependencies: REQ-1, REQ-2

4.7 System Feature 7

Title: Automated Stub / Driver Generation.

Priority: Low.

Description: The system will generate stub and driver methods / classes / modules if the user wish to have them for testing purpose.

Functional Requirement: REQ-7

Dependencies: REQ-1, REQ-2, REQ-4

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The software should manage the work load properly. It should have a very good memory management system. It should not create any kind of disturbance / lag when the user is writing code or using the primary features of this software.

5.2 Security Requirements

Security Requirements: The software will provide password protected security feature for user's projects. So, that important project files could be accessed by unauthorized person.

5.3 Software Quality Attributes

The system should be reliable in a sense that, sudden system failure should not damage user files. The system will automatically save a copy of user's working files after 2 minutes.

It should be easily maintainable and extendable to cope with the frequent changes that occur in programming languages.

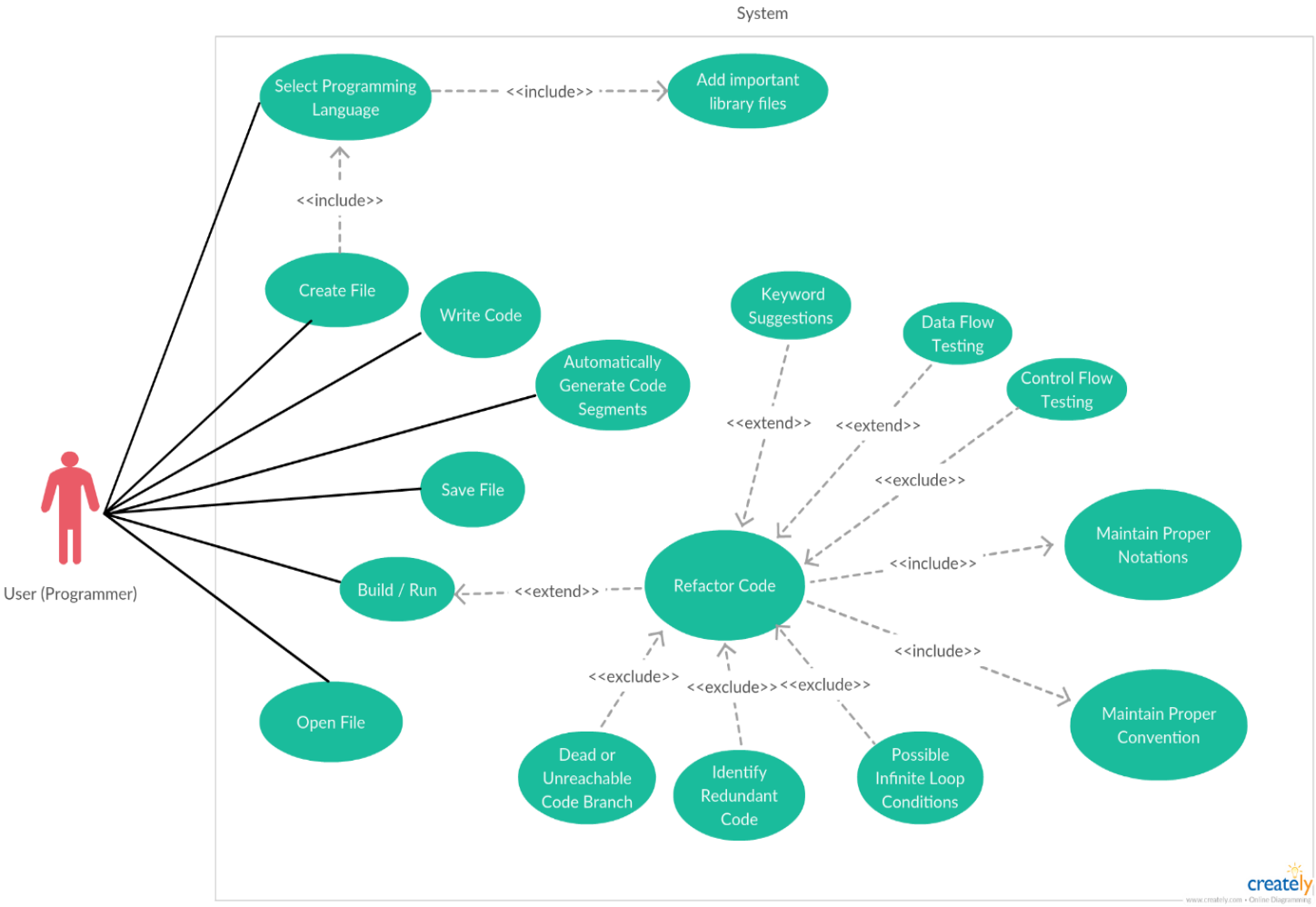
Test environments should be built for the system to allow testing of its different functions.

Appendix A: Glossary

Word	Meaning
Refactoring	Re-organizing
Compile	Convert source codes to machine executable code
Programming Convention	The rules that should be followed while writing code
Data Flow Anomaly	Unexpected error in data flow of a source program
Control Flow Anomaly	Unexpected error in control flow of a source program
Stub	Empty function / class / package
Driver	Empty function / class / package
Redundant	Unnecessary line of code
Infinite Loop	A loop process that runs endlessly

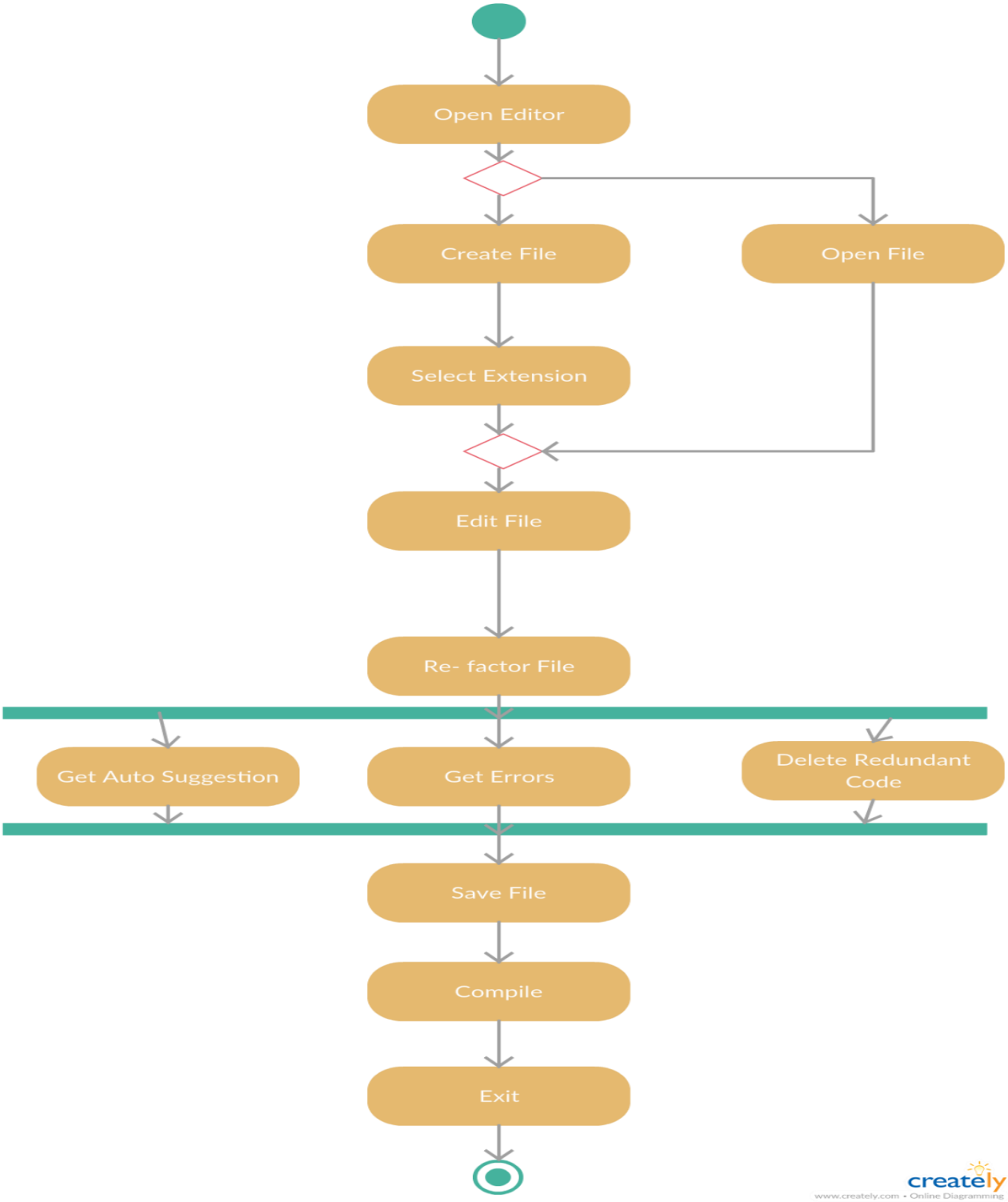
Appendix B: Analysis Models

a. Use Case Diagram:



The general user of this software will be professional developers, testers, teachers and students. The users can create new file or open any other existing files. They can write code and save them for future use. Some added functionalities are provided for coding flexibility of the users. They users will get real-time syntax suggestions as well as repeated parts of the code will also be generated automatically. The refactoring functionality of the system allows the user to improve their code's quality and make them easily manageable and readable. The users can also compile their codes using built in compilers to convert their source code to executable program file.

b. Activity Diagram:



Appendix C: To Be Determined List

1. The user interface (UI) design will be prototyped and finalized after all the requirements are checked and verified.
2. Support for Linux and other operating systems will be developed after the initial release of “ReShaper – Code Refactoring Tool V-1.0” for windows.